Proceedings of the International Congress of Mathematicians
Berkeley, California, USA, 1986

Elliptic Curves and Number-Theoretic Algorithms

H. W. LENSTRA, JR.

1. Introduction. In this lecture we shall discuss a problem that has fascinated many mathematicians throughout history, such as Eratosthenes ($\sim -284-\sim -202$), Fibonacci ($\sim 1180-\sim 1250$), Fermat (1601–1665), Euler (1707–1783), Legendre (1752–1833), and Gauss (1777–1855). This is the problem of how to find the *prime factor decomposition* of a given large integer.

Surveys of methods that are used for this purpose can be found in Riesel's recent book [**27**] and in the contributions to [**21**]. The present lecture is devoted to a development that took place since the appearance of Riesel's book, namely, the introduction of *elliptic curves.*

Two stages can be distinguished in most methods to find the prime factorization of a given number. In the first stage (*primality testing*) one decides whether the number is prime or composite. In the second stage (*factorization*) one finds a nontrivial divisor of the number, if it is composite. It is clear that the complete prime factor decomposition can be obtained by applying a primality testing algorithm and a factorization algorithm recursively. Elliptic curves can be applied both to primality testing and to factorization, and they give rise to algorithms with an excellent performance, both in theory and in practice.

Primality testing is considered to be easier than factorization. Suppose, for example, that two 100-digit numbers p and q have been proved prime; this is easily within reach of the current primality testing methods. Suppose moreover that the numbers p and q are thrown away by mistake, but that the product pq is saved. How to recover p and q? It must be felt as a defeat for mathematics that, in these circumstances, the most promising approaches are searching the waste paper basket and applying mnemo-hypnotic techniques.

Until recently, the subject of primality testing and factorization was not taken seriously by most mathematicians. Nowadays, a change in this attitude is noticeable. Partly, this change is due to the introduction of more sophisticated mathematical techniques than were used before. Indeed, the use of elliptic curves, which is the main topic of this lecture, has been referred to as the first application of twentieth-century mathematics to the problem of prime factor decomposition.

Another reason for the increased interest in this area is the possibility to apply number theory to the outside world. The existence and uniqueness of the prime factor decomposition constitute the *fundamental theorem of arithmetic*, and this theorem plays indeed a basic role. For example, a number-theoretical question about a positive integer n—can n be written as the sum of two squares? what is the order of the multiplicative group $(\mathbf{Z}/n\mathbf{Z})^*$?—is considered as settled if it is answered in terms of the prime factorization of n. Given the basic role of the prime factor decomposition in number theory, it seems reasonable to suppose that algorithms to achieve this prime factor decomposition play an important role in possible applications of number theory. To date, the most striking illustration is the cryptographic scheme devised by Rivest, Shamir, and Adleman [**28**]. For the use of this scheme it is essential that primality testing is *easy*, and the security of the system depends on the fact that factorization is *hard*. It should be remarked that, to a certain extent, this is *negative* application: if a better factoring method is discovered, then the application may cease to exist. This remark should serve as a stimulus for those mathematicians to whom the possibility of applying number theory to the outside world does not appeal and who wish to restore the purity of their science.

To test a given integer $n > 1$ for primality, one usually subjects it to a series of *pseudoprime tests*. Most of these tests are based on a variant of *Fermat's theorem*. This theorem asserts that if n is prime then $a^n \equiv a \bmod n$ for all integers a. These pseudoprime tests have the property that any prime number passes them, but that a composite number is very unlikely to pass them. Hence a single test that n fails to pass suffices to prove that n is composite, although it does not readily yield a factor of n. If, on the other hand, n passes many pseudoprime tests, then it is very likely that n is a prime number. The problem then becomes how to *prove* that n is a prime number. It may be said that the real difficulty of primality testing algorithms is not to *obtain* the answer, "prime" or "composite," but to prove the *correctness* of the answer, in the case it is "prime." For this reason one sometimes speaks about primality *proving* algorithms.

If a primality test decides that a number is *not* prime then, as we just noted, it usually does not exhibit a factor of the number. To obtain a factor one applies a factorization algorithm. In contrast to primality testing, the difficulty of factorization is to *obtain* the answer, i.e., a nontrivial divisor of the number; checking the correctness of the answer, once it is obtained, is completely trivial. The total freedom one has in the choice of the method by which to obtain a nontrivial divisor seems to be one of the reasons that there is much more variety in factorization algorithms than in primality tests. Indeed, it is not a priori clear why methods that depend on a mathematical theory would be better than nonmathematical methods, and why factorization should be beyond the abilities of competent clairvoyants or religious officers.

The elliptic curve methods that form the subject of this lecture are best understood as analogue of certain older algorithms, which are discussed in §2. These older algorithms depend on properties of the *multiplicative group*, in particular

on the fact that for a prime number p the order of the multiplicative group $(\mathbf{Z}/p\mathbf{Z})^*$ equals $p-1$. We remark that the algorithms discussed in §2 are by no means the best algorithms that were used before elliptic curves were introduced; we only discuss them because they are helpful in motivating and understanding the new methods.

Section 3 contains the basic properties of elliptic curves that we need. The best reference is Silverman's recent textbook [**35**]. As most of the literature on the subject, this book restricts itself to elliptic curves that are defined over *fields*. For our purposes it is more natural, both from a conceptual and from an expository point of view, to work with elliptic curves that are defined over *rings*. The general theory of elliptic curves over commutative rings with 1 can be found in [**16**, Chapter 2]. In §3 we give the basic definitions, but only in the case that the ring in question satisfies a certain condition; this condition is satisfied, for example, if the ring is a field, and also if the ring is *finite*, which is the case in our applications. This condition allows us to give a very straightforward definition: an elliptic curve is defined by a ternary homogeneous cubic polynomial of a certain normal form; to keep this normal form as simple as possible we assume that 6 is a unit of the ring. The set of points of the curve over the ring is then defined as the set of zeros of this polynomial in a suitably defined projective plane. It is a basic property of elliptic curves that this set of points has the structure of an *abelian group*. It should be remarked that in principle it is possible, by more or less artificial considerations, to avoid elliptic curves over rings that are not fields in the description and analysis of the algorithms that we shall discuss. This was, in fact, done in the original publications [**30**, **20**, **14**].

We mentioned above that a number of older primality testing and factorization methods depend on the fact that the order of the multiplicative group $(\mathbf{Z}/p\mathbf{Z})^*$ modulo a prime number p equals $p-1$. Likewise, in the elliptic curve methods an important role is played by the order of the group $E(\mathbf{Z}/p\mathbf{Z})$ of points of an elliptic curve E over $\mathbf{Z}/p\mathbf{Z}$, for a prime number p. By a theorem of Hasse from 1934, this order is of the form $p+1-t$, where t is an integer depending on E and p for which $|t| \leq 2\sqrt{p}$. It may be said that the success of the new methods is due to the fact that, for fixed p, this number t varies if one varies the elliptic curve E. In §4 we discuss several methods to calculate the number t.

In §5 it is explained how to do primality testing with the help of elliptic curves. In particular, we discuss the algorithms of Goldwasser-Kilian [**14**] and Atkin [**2**]. Atkin's method is of great practical value, and on most numbers on which it has been tried it is much faster than the previous champion, which is the Cohen-Lenstra version of the test of Adleman, Pomerance, and Rumely [**1**, **9**, **10**].

Section 6, finally, describes the elliptic curve factorization method [**20**]. It is, at the moment, the undisputed champion among factoring methods for the great majority of numbers. The quadratic sieve algorithm of Pomerance [**26**], which was the previous champion, still seems to perform better on numbers that are built up from two primes of the same order of magnitude. The elliptic curve

method has the very attractive property that its speed depends on the size of the smallest prime divisor of the number n that is being factored: smaller prime factors are easier to find. The quadratic sieve and many other fast factoring algorithms do not have this property; they have a running time that only depends on the size of n and not on the size of its prime factors.

By $\mathbf{F}_q$ we shall denote a finite field of cardinality q. Rings are supposed to be commutative with a unit element, and the latter is supposed to be preserved by ring homomorphisms. The group of units of a ring R is denoted by R^*.

2. Multiplicative methods. In this section we discuss two older algorithms for primality testing and factorization, which depend on properties of the multiplicative group. In practice, these algorithms are not feasible for all numbers, but only if certain conditions are satisfied.

We begin with primality testing. The following theorem is due to Pocklington [**24**].

THEOREM 1. *Let n be an integer, $n > 1$, and s a positive integer dividing $n-1$. 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 prime divisor p of n is $1 \bmod s$, and if $s > \sqrt{n} - 1$ then n is prime.

The proof is as follows. Let p be a prime divisor of n, and write $b = (a^{(n-1)/s} \bmod n)$. From $a^{n-1} \equiv 1 \bmod n$ it follows that $b^s \equiv 1 \bmod p$, so the order of $(b \bmod p)$ in the group $\mathbf{F}_p^*$ divides s. Also, if q is a prime divisor of s, then $b^{s/q}$ is not $1 \bmod p$, since by hypothesis $a^{(n-1)/q} - 1$ is not divisible by p. Therefore the order of $(b \bmod p)$ is not a divisor of s/q, for any prime q dividing s, so this order is equal to s itself. By Lagrange's theorem in group theory it follows that s divides $\#\mathbf{F}_p^* = p-1$. This proves the first assertion of the theorem. If also $s > \sqrt{n} - 1$ then it follows that $p > \sqrt{n}$, and this can only be true for all primes p dividing n if n is prime. This proves Theorem 1.

The use of Theorem 1 in primality testing is as follows. Let n be an integer > 1 that one believes to be prime, for example because it passes pseudoprime tests as described in [**17**, p. 379; **27**, p. 98]. Denote by s the largest divisor of $n-1$ that one is able to factor completely into primes, and suppose that $s > \sqrt{n} - 1$. Now pick a random nonzero integer $a \pmod n$, and test whether it satisfies the two conditions of Theorem 1. Observe that these conditions are easy to test: the prime divisors q of s are known, the powers $a^{n-1} \pmod n$ and $a^{(n-1)/q} \pmod n$ can be calculated with $O(\log n)$ multiplications and squarings $\bmod n$, and the greatest common divisors can be calculated by means of the Euclidean algorithm. If all conditions are found to be satisfied then it follows from the theorem that n is indeed prime, as required.

It should be mentioned that if n is prime it should not be difficult to find an element $a \in \mathbf{Z}/n\mathbf{Z}$ satisfying the conditions of the theorem. Clearly, any nonzero

$a \in \mathbf{Z}/n\mathbf{Z}$ must satisfy the first condition, if n is prime. It is easy to show that, for fixed q, the second condition is satisfied with probability $1 - q^{-1}$, if n is a given prime and $a \neq 0$ is drawn at random. The probability that a satisfies the second condition for all q may be somewhat smaller, but in any case it is at least $c_0/\log\log n$ for some positive constant c_0; also, it is not difficult to prove a slightly more general version of the theorem, in which a is allowed to depend on q.

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 is able to factor completely. This is the case if $n-1$ has many small prime factors, which happens, for example, for the Fermat numbers $n = 2^k + 1$. Theorem 1 is also useful 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.

There is an analogue to Theorem 1 with the multiplicative group replaced by a *twisted* multiplicative group. For example, if p is prime then the group $\mathbf{F}_{p^2}^*/\mathbf{F}_p^*$ is a twisted multiplicative group, and it has order $(p^2-1)/(p-1) = p+1$. This leads to primality tests that can be used for numbers n for which $n+1$ has a large completely factored divisor. This is the case, for example, for the Mersenne numbers $n = 2^k - 1$. These tests are classically formulated in terms of *Lucas sequences.*

We refer to [**27**, **38**] for the details of these and other generalizations of Theorem 1, and for a description of the primality tests that are based on a combination of the $(n-1)$- and $(n+1)$-methods. If n has the property that at least one of $n \pm 1$ can be written as the product of a completely factored number and a prime number q that, recursively, has the same property, then the primality of n can be proved by repeated application of the two methods. This method was developed by Selfridge and Wunderlich [**32**], and they found empirically that it can be applied to most primes of at most 35 digits, if "completely factored" is taken to mean "built up from primes below 30030." The generalizations due to Williams et al. [**38**] can be used for most prime numbers of at most 80 digits.

The advantage of elliptic curves in this context is that there are so many of them. Each elliptic curve gives rise to a group, and the order of this group varies with the curve. Instead of using the numbers $n \pm 1$, one uses essentially a random number in the neighborhood of n, and one can keep changing the curve until this number factors in the desired way. We refer to §5 for more details.

Next we consider a factorization method that also depends on the multiplicative group. It was invented by Pollard [**25**], and it is known as the *Pollard* $(p-1)$*-method.*

The Pollard $(p-1)$-method attempts to find a nontrivial divisor of a composite integer $n > 1$ in the following way. Pick $a \in \mathbf{Z}/n\mathbf{Z}$ at random, and select a positive integer k that is divisible by many small prime powers; for example, one can take $k = \operatorname{lcm}\{1, 2, \ldots, w\}$ for a suitable bound w. Next one calculates

$a_k = (a^k \bmod n)$. This can be done by performing $O(\log k)$ squarings and multiplications $(\bmod n)$. Finally, one calculates $\gcd(a_k - 1, n)$ by means of Euclid's algorithm, and one hopes that this gcd is a nontrivial divisor of n.

Pollard's $(p-1)$-method is usually successful if n has a prime divisor p for which $p-1$ is built up from small prime factors only. Suppose, to be specific, that $p-1$ divides k, and that p does not divide a. Since the order of $(\mathbf{Z}/p\mathbf{Z})^*$ equals $p-1$, it then follows that $a^k \equiv 1 \bmod p$, so p divides $\gcd(a_k - 1, n)$. In many cases one has $p = \gcd(a_k - 1, n)$, and the method finds a nontrivial 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.

We refer to [**25**] for a refinement of the method, which improves its practical performance; to [**39**] for a variant that uses a twisted multiplicative group, and for which $p+1$ rather than $p-1$ should be built up from small prime factors; and to [**3**] for a generalization that appears to be only of theoretical value.

The advantage of elliptic curves is the same as with primality testing. If one uses an elliptic curve rather than the multiplicative group, then $p \pm 1$ is replaced by a number in the neighborhood of p that varies with the curve, and one can keep changing the curve until the algorithm is successful; one may hope that a fair proportion of the numbers in the neighborhood of p is built up from small primes only, so that not too many curves need be tried. More details can be found in §6.

3. Elliptic curves over rings. Let R be a ring. A finite collection $(a_i)_{i \in I}$ of elements of R will be called *primitive* if it generates R as an R-ideal, i.e., if there exist $b_i \in R$, for $i \in I$, such that $\sum_{i \in I} b_i a_i = 1$. This terminology will in particular be applied to *vectors* and to *matrices* that have coefficients in R. Notice that if R is a field, a collection $(a_i)_{i \in I}$ is primitive if and only if not all a_i are zero.

In the sequel we assume that R satisfies the following two conditions:

(i) $6 \in R^*$;

(ii) for all positive integers n, m and every primitive matrix $(a_{ij})_{1 \le i \le n,\ 1 \le j \le m}$ over R with the property that all 2×2-subdeterminants vanish ($a_{ij}a_{kl} - a_{il}a_{kj} = 0$ for all i, j, k, l with $1 \le i < k \le n,\ 1 \le j < l \le m$) there exists an R-linear combination of the rows that is primitive as an element of R^m.

If R is a field the first condition means that $\operatorname{char} R \neq 2, 3$. We impose this condition only to simplify the exposition; for $6 \notin R^*$ one must work with more general normal forms for elliptic curves, as in [**35**, Chapter 3].

The second condition, however, is essential for the definition of elliptic curves and their addition law that we shall give. Condition (ii) means that every *projective R-module* of rank one is free, or equivalently that the *Picard group* $\operatorname{Pic} R$ of R vanishes [**4**]. Obviously, the condition is satisfied for fields, and below

we shall see that it is also satisfied for finite rings. More generally, it holds for rings that have only finitely many maximal ideals. If R is a Dedekind ring, for example, the ring of integers in a number field, then (ii) is true if and only if the class group of R is trivial.

It is easy to prove that the primitive element of R^m whose existence is postulated by (ii) is in fact uniquely determined up to multiplication by units.

Let R be a ring satisfying (i) and (ii). The unit group R^* acts on the set of primitive triples $(x, y, z) \in R^3$ by $u(x, y, z) = (ux, uy, uz)$. The set of orbits under this action is denoted by $\mathbf{P}^2(R)$, and called the *projective plane* over R. The orbit of (x, y, z) is denoted by $(x : y : z)$.

An *elliptic curve* over R is a pair of elements $a, b \in R$ for which $4a^3 + 27b^2 \in R^*$. These elements are to be thought of as the coefficients in the homogeneous Weierstrass equation

$$y^2 z = x^3 + axz^2 + bz^3.$$

We denote the elliptic curve (a, b) by $E_{a,b}$, or simply by E. If we multiply the above equation by u^6, for some $u \in R^*$, and replace $u^2 x, u^3 y$ by x, y, respectively, then we obtain the equation for $E_{a',b'}$, where $a' = u^4 a$ and $b' = u^6 b$. Two such curves are said to be *isomorphic* over R.

Let $E = E_{a,b}$ be an elliptic curve over R. The *set of points* $E(R)$ of E over R is defined by

$$E(R) = \{(x : y : z) \in \mathbf{P}^2(R) \colon y^2 z = x^3 + axz^2 + bz^3\}.$$

The point $(0 : 1 : 0) \in E(R)$ is called the *zero point* of the curve, and denoted by O. Notice that if R is a field this is the only element of $E(R)$ whose z-coordinate is zero.

It is a basic fact that $E(R)$ has in a natural way the structure of an *abelian group* with O as the neutral element. The group law, which is written additively, is such that $-(x : y : z) = (x : -y : z)$ for all $(x : y : z) \in E(R)$. To define the group law we first consider the case that R is a *field*. In this case the addition formulae, and the proof that $E(R)$ is a group, can be found in [**35**, Chapter 3]. We briefly summarize what we need.

Let R be a field, and let $P_1, P_2 \in E(R)$. To add P_1 and P_2, consider the straight line passing through P_1 and P_2 (the tangent line to the curve if $P_1 = P_2$). The line and the curve have three intersection points, if we count them with suitable multiplicities, and two of them are P_1 and P_2. If Q is the third one, then $P_1 + P_2 = -Q$. To turn this geometric description into algebraic formulae, we may suppose that P_1 and P_2 are nonzero and that $P_1 \neq -P_2$. Then we can write $P_i = (x_i : y_i : 1)$ for $i = 1, 2$, where (x_i, y_i) lie on the affine curve $y^2 = x^3 + ax + b$. The straight line is given by $y = \lambda x + \nu$, where

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \quad \text{or} \quad \lambda = \frac{x_2^2 + x_2 x_1 + x_1^2 + a}{y_2 + y_1}$$

and $\nu = y_1 - \lambda x_1$. Notice that $P_1 \neq -P_2$ implies that at least one of the values for λ is well defined, and that they are equal if they are both well defined. The

sum $P_3 = P_1 + P_2$ is now given by $P_3 = (x_3 : y_3 : 1)$, where

$$x_3 = \lambda^2 - x_1 - x_2, \qquad y_3 = -(\lambda x_3 + \nu).$$

This gives the addition formulae if R is a field, but for the sequel it is desirable to bring them into homogeneous form. To do this, one replaces x_i and y_i by x_i/z_i and y_i/z_i, respectively, and one clears the denominators. Then one finds that the sum of two points $P_1 = (x_1 : y_1 : z_1)$, $P_2 = (x_2 : y_2 : z_2)$ on $E(R)$ is given by one of two formulae $(q_1 : r_1 : s_1), (q_2 : r_2 : s_2)$, depending on which formula for λ is used. Here $q_1, \ldots, s_2$ are certain polynomial expressions in $x_1, y_1, z_1, x_2, y_2, z_2, a$ with integer coefficients. It turns out that for every pair $(P_1, P_2) \in E(R) \times E(R)$ except $(P_1, P_2) = (O, O)$ at least one of these two formulae is meaningful in the sense that it does not give $(0 : 0 : 0)$, and that any of the two that is meaningful actually gives the sum of P_1 and P_2 in the group $E(R)$. For the remaining pair (O, O) we know of course that $O + O = O = (0 : 1 : 0)$, but this formula is not satisfactory because it does *not* have the property of correctly giving the sum $P_1 + P_2$ for all pairs of points P_1, P_2 for which it is meaningful. To remedy this situation one has to develop an addition law that is valid "in a neighborhood of (O, O)," and that can be done as in [**35**, Chapter IV, §1]. The result is that one finds nine polynomial expressions q_i, r_i, s_i $(i = 1, 2, 3)$ in $x_1, y_1, z_1, x_2, y_2, z_2, a, b$ with integer coefficients, with the property that the sum of *any* two points $P_1 = (x_1 : y_1 : z_1)$, $P_2 = (x_2 : y_2 : z_2)$ on $E(R)$ is given by one of the three formulae $(q_i : r_i : s_i)$, $i = 1, 2, 3$, and that in fact any of the three formulae that is meaningful is correct. The latter statement is equivalent to nine formal identities $q_1 r_2 - q_2 r_1 = 0, \ldots, r_2 s_3 - r_3 s_2 = 0$ in the ring $\mathbf{Z}[a, b, x_1, y_1, z_1, x_2, y_2, z_2]/I$, where $a, \ldots, z_2$ are considered as polynomial variables and I denotes the ideal generated by the two polynomials $y_i^2 z_i - x_i^3 - a x_i z_i^2 - b z_i^3$, $i = 1, 2$. Likewise, the fact that $P_1 + P_2$ lies again on the curve, and that the addition defined in this way satisfies the group axioms, with the zero element and the negatives of points as indicated above, is expressed by a series of formal identities in the same ring. Nine explicit polynomials $q_1, \ldots, s_3$ with all these properties can be found in [**19**].

We now drop the condition that R be a field. To add two points $P_1 = (x_1 : y_1 : z_1)$, $P_2 = (x_2 : y_2 : z_2)$ on $E(R)$ one proceeds as follows. One uses the same nine polynomial expressions that appeared above to obtain a 3×3-matrix

$$\begin{pmatrix} q_1 & r_1 & s_1 \\ q_2 & r_2 & s_2 \\ q_3 & r_3 & s_3 \end{pmatrix}$$

with entries from R. This is a primitive matrix, since otherwise there would be a maximal ideal $\mathfrak{m} \subset R$ containing all nine entries; but this would contradict the fact that at least one of the rows can be used to add the two points $P_1 \bmod \mathfrak{m}, P_2 \bmod \mathfrak{m}$ on the elliptic curve $E_{a \bmod \mathfrak{m}, b \bmod \mathfrak{m}}(R/\mathfrak{m})$ over the field $R/\mathfrak{m}$. Also, all 2×2-subdeterminants of the matrix are zero, so by condition (ii) above there is an R-linear combination (q_0, r_0, s_0) of the rows that is primitive;

moreover, the orbit of (q_0, r_0, s_0) under R^* is uniquely determined. We now define the sum of P_1 and P_2 on $E(R)$ to be $(q_0 : r_0 : s_0)$.

The fact that $E(R)$ is closed under this operation, and that the addition defined in this way satisfies the group axioms, with the zero element and the negatives of points as indicated earlier, is a consequence of the formal identities that we mentioned above. We omit the details, which are somewhat tedious.

It is a natural question to ask for an *algorithm* to add two points on $E(R)$. From the definition of addition we see immediately that, given the formulae from [**19**], it suffices to have an algorithmic version of condition (ii): one needs a method to *find* the primitive linear combination that is asserted to exist. Before we describe such a method for the case that R is *finite* it should be pointed out that at the moment this method has only theoretical value. Namely, for the purposes that we have in mind (see the following sections) there is a much easier method, as follows. Pick any nonzero entry from the matrix, and determine whether it is a unit in R. If it is, then the row containing that element is primitive, and one is done. If it isn't, then one knows a nonzero nonunit of R, and in each of the cases that we shall consider this is also satisfactory. Suppose for example, that $R = \mathbf{Z}/n\mathbf{Z}$, where n is an integer that one is trying to factor; then a nonzero nonunit of R leads to a nontrivial divisor of n, which is exactly what one wants.

Assume now that R is a *finite* ring. We assume that the elements of R are represented by elements of a certain finite set S; one may think of S, for example, as consisting of strings of zeros and ones. It is allowed that two distinct elements s, s' of S represent the same element of R, but we do require that given $s, s' \in S$ there is an efficient algorithm to decide whether this is the case. Here "efficient" may be taken to mean that the time needed by the algorithm is bounded by a polynomial function of $\log \#S$. We also require that there is an efficient algorithm to do *addition* in R; that is, given $s, s' \in S$, one should be able to find an element of S that represents the sum of the elements represented by s and s'. Likewise we require that *subtraction* and *multiplication* can be done efficiently, as well as the solution of equations of the sort $cx = d$ (given c and d, find x), if they are solvable. Finally we require that an element representing $1 \in R$ is known.

With these hypotheses there is an efficient algorithm that given a primitive $n \times m$-matrix (a_{ij}) as in condition (ii) produces a linear combination of the rows that is primitive; here "efficient" means that the time needed by the algorithm is bounded by a polynomial function of n, m, and $\log \#S$. We begin with a lemma.

LEMMA. *Let R, S be as above, and denote by t the least positive integer for which $2^{t+1} > \#S$. Then for every $c \in R$ there exists $x \in R$ with $c^{t+1}x = c^t$. Moreover, an element $c \in R$ is nilpotent if and only if $c^t = 0$.*

PROOF. Consider the sequence of ideals

$$R \supset Rc \supset Rc^2 \supset \cdots \supset Rc^t \supset Rc^{t+1}.$$

If any two consecutive ideals in this chain are distinct, one obtains $\#S \geq \#R \geq \text{index}[R: Rc^{t+1}] \geq 2^{t+1}$, which is a contradiction. Hence $c^i = c^{i+1}x$ for some $x \in R$ and some integer i with $0 \leq i \leq t$, and the first statement of the lemma follows upon multiplication by c^{t-i}.

If u is an integer with $u > t$, then it follows that $c^u x = c^{u-1}$. Therefore, if c is nilpotent, the smallest integer u with $c^u = 0$ cannot be larger than t. This implies the last statement of the lemma.

It follows from the lemma that there is an efficient algorithm to decide whether an element of the ring is nilpotent.

We now describe an efficient algorithm that given an $n \times m$-matrix $A = (a_{ij})$ as in (ii) finds a primitive combination of its rows. The algorithm proceeds by recursion on the cardinality of R. If R is the zero ring (which can be decided by testing whether $1 = 0$, where $0 = 1 - 1$), then any row of the matrix is primitive. Now suppose that R is not the zero ring. Since the matrix is primitive, not all of its entries are nilpotent. Let c be an entry that is not nilpotent. Using the lemma, solve $c^{t+1}x = c^t$. Then $c^{2t}x^t = c^t$, so if we put $e = c^t x^t$ then e is an idempotent: $e^2 = e$. Also, from $c^t e = c^t \neq 0$ one sees that $e \neq 0$. If now $e = 1$ then c is a unit, so the row of the matrix containing c is primitive, and one is done. Suppose therefore that $e \neq 1$. Then $R_1 = Re$ and $R_2 = R(1-e)$ are nonzero commutative rings with unit elements e and $1 - e$, respectively. Moreover, the map $R \to R_1 \times R_2$ sending $r \in R$ to $(re, r(1-e))$ is an isomorphism of rings. The matrix A gives rise to a matrix A_1 over R_1 and a matrix A_2 over R_2. Now notice that, for each $i = 1, 2$, the map $S \to R \to R_i$ shows that the set S can again be used to represent the elements of R_i, and that the same conditions as for R are satisfied. Hence, recursively, we can find an R_i-linear combination of the rows of A_i that is primitive as an element of R_i^m, for each $i = 1, 2$. Adding these two rows in R^m one finds the desired primitive linear combination of the rows of A. This finishes the description of the algorithm.

We remark that, in the above algorithm, the element $c \in R$ is mapped to an element $(c_1, c_2) \in R_1 \times R_2$ for which c_1 is a unit and c_2 is nilpotent. Hence the row of A_1 containing c_1 is already primitive, and the recursion is only needed for the ring R_2. Since the number of nilpotent entries in A_2 is at least one more than in the matrix A, this shows that the depth of the recursion is bounded by nm. In the case that is of interest to us one has $nm = 9$.

4. The number of points on an elliptic curve. Let R be a *finite* ring with $6 \in R^*$, and $E = E_{a,b}$ an elliptic curve over R. In this section we discuss the order of the finite group $E(R)$.

If $f: R \to R'$ is any ring homomorphism from R to a ring R' that also satisfies the two conditions (i), (ii) from §3, then $E_{f(a),f(b)}$ is an elliptic curve over R'. We denote this elliptic curve again by E.

If R contains an element c that is neither a unit nor nilpotent then, as we saw in the previous section, R can be written as the product of two nonzero rings. By induction on $\#R$ it follows that R is isomorphic to the product of

finitely many rings R_i, where each R_i is such that every element of R_i is either nilpotent or a unit. Then each R_i is a *local* ring, which means that the set $\mathfrak{m}_i$ of nonunits of R_i forms an ideal of R_i; this ideal must be maximal, so that $R_i/\mathfrak{m}_i$ is a field. It is now easy to see that $E(R)$ is isomorphic to the product of the groups $E(R_i)$, so that $\#E(R) = \prod_i \#E(R_i)$. Furthermore, from Hensel's lemma one can deduce that for each i the natural group homomorphism $E(R_i) \to E(R_i/\mathfrak{m}_i)$ is *surjective* and that its kernel has the same cardinality as $\mathfrak{m}_i$, so that $\#E(R_i) = \#E(R_i/\mathfrak{m}_i) \cdot \#\mathfrak{m}_i$. Summarizing, we have

$$\frac{\#E(R)}{\#R} = \prod_{\mathfrak{m}} \frac{\#E(R/\mathfrak{m})}{\#R/\mathfrak{m}},$$

where $\mathfrak{m}$ ranges over the set of maximal ideals of R. If these maximal ideals are known, then this formula reduces the computation of $\#E(R)$ to the case that R is a field. If $R = \mathbf{Z}/n\mathbf{Z}$ for some positive integer n, then the above formula reads

$$\frac{\#E(\mathbf{Z}/n\mathbf{Z})}{n} = \prod_{p} \frac{\#E(\mathbf{F}_p)}{p},$$

where p ranges over the set of primes dividing n. Notice that the same formula holds with the order of the elliptic curve replaced by the Euler ϕ-function, which is the order of the multiplicative group.

Assume, for the rest of this section, that R is a finite *field*, of characteristic different from 2 and 3. Denote the cardinality of R by q, so that we may write $R = \mathbf{F}_q$. We assume that an explicit representation for the elements of R is available, as in the previous section, and that each arithmetic operation in R can be performed in time $O((\log q)^2)$.

According to a theorem of Hasse (1934) we have $\#E(\mathbf{F}_q) = q + 1 - t$, where t is an integer satisfying $|t| \leq 2\sqrt{q}$. Four methods have been proposed to calculate the number $\#E(\mathbf{F}_q)$ or, equivalently, the number t.

The first method, which was employed by Lang and Trotter [**18**], depends on the formula

$$\#E(\mathbf{F}_q) = 1 + \sum_{x \in \mathbf{F}_q} (1 + \chi(x)),$$

where $\chi(x)$ denotes the element of $\{0, 1, -1\}$ that maps to $(x^3 + ax + b)^{(q-1)/2}$ under the natural map $\mathbf{Z} \to \mathbf{F}_q$. To prove this formula one simply notes that, for fixed $x \in \mathbf{F}_q$, the number of $y \in \mathbf{F}_q$ with $y^2 = x^3 + ax + b$ is given by $1 + \chi(x)$. Applying this formula in a straightforward way leads to an algorithm to calculate $\#E(\mathbf{F}_q)$ that takes time $O(q^{1+\varepsilon})$, for any $\varepsilon > 0$.

The second method, which is significantly faster, is *probabilistic* in the sense that it depends on random choices. It is analogous to an algorithm of Shanks [**33**] for the calculation of class numbers of imaginary quadratic fields. We give a brief description.

First, one picks a random point $P \in E(\mathbf{F}_q)$. This is done by selecting random elements $x \in \mathbf{F}_q$ until an element is found for which $x^3 + ax + b$ is a square in $\mathbf{F}_q$; this can be tested by checking whether $\chi(x) \neq -1$, with χ as above. If such

an x has been found, one can find an element $y \in \mathbf{F}_q$ with $y^2 = x^3 + ax + b$ by applying another probabilistic algorithm of Shanks [**34**] or by applying a general zero-finding routine for polynomials over finite fields [**17**, §4.6.2]. The point $P = (x : y : 1)$ is now on the curve.

Next one determines all integers m for which both $|m - (q+1)| \leq 2\sqrt{q}$ and $m \cdot P = O$. Clearly such integers exist, since $m = \#E(\mathbf{F}_q)$ has these properties. By means of the "baby step–giant step" strategy, for the details of which we refer to [**33**], all these integers m can be found in time $O(q^{(1/4)+\varepsilon})$, for any $\varepsilon > 0$.

If m is unique, then $m = \#E(\mathbf{F}_q)$, and one is done. If m is not unique, then the difference between any two consecutive m's equals the order of P, and it is easy to see that P cannot generate the group $E(\mathbf{F}_q)$, if $q \geq 37$. In the latter case one selects another random point $P' \in E(\mathbf{F}_q)$, and in a similar way one determines the order of the point P' modulo the subgroup generated by P. In this way one continues until the order k of the subgroup that has been found satisfies $|k - (q+1)| \leq 2\sqrt{q}$. Then $\#E(\mathbf{F}_q) = k$, if $q \geq 37$.

This algorithm has expected running time $O(q^{(1/4)+\varepsilon})$, for any $\varepsilon > 0$, and it determines not only the order of $E(\mathbf{F}_q)$ but also its group structure. It is of practical value if q has not more than approximately 20 decimal digits.

The third method that we discuss is due to Schoof [**30**]. It is completely deterministic. The method depends on properties of the *Frobenius endomorphism* ϕ of the curve, which is defined as follows. Denote by K an algebraic closure of $\mathbf{F}_q$. Then ϕ is the automorphism of the abelian group $E(K)$ defined by

$$\phi(x : y : z) = (x^q : y^q : z^q).$$

Notice that $E(\mathbf{F}_q)$ may be considered as a subgroup of $E(K)$, and that $E(\mathbf{F}_q) = \{P \in E(K) : \phi(P) = P\}$. It is a basic theorem that ϕ satisfies the quadratic equation $\phi^2 - t\phi + q = 0$ in the endomorphism ring of $E(K)$, where t is the integer for which $\#E(\mathbf{F}_q) = q + 1 - t$.

To determine t one now observes that it suffices to determine $t \bmod l$ for all odd primes $l \leq c_1 \log q$ that are different from char $\mathbf{F}_q$; here c_1 is a positive constant, chosen such that $\prod l > 4\sqrt{q}$ for all q. Namely, if one knows all these $t \bmod l$ then one can determine $t \bmod \prod l$ by means of the Chinese remainder theorem, and since $|t| \leq 2\sqrt{q}$ this suffices to find t and hence $\#E(\mathbf{F}_q)$.

Now let l be an odd prime number, $l \neq \operatorname{char} \mathbf{F}_q$. To determine $t \bmod l$, one first calculates the polynomial ψ_l defined by

$$\psi_l = l \cdot \prod (X - x),$$

with x ranging over the set of those elements of K for which there exists $y \in K$ for which $(x : y : 1)$ is an element of $E(K)$ of order l. It is known that ψ_l has degree $(l^2 - 1)/2$ and belongs to $\mathbf{F}_q[X]$. The polynomial ψ_l can be calculated by means of recursion formulae that can be found, for example, in [**35**, Chapter III, Exercise 3.7].

Define the ring T by

$$T = \mathbf{F}_q[X, Y]/(\psi_l, Y^2 - X^3 - aX - b).$$

Every element of T has a unique representation

$$\sum_{i=0}^{(l^2-3)/2} \sum_{j=0}^{1} a_{ij} \overline{X}^i \overline{Y}^j \quad \text{with } a_{ij} \in \mathbf{F}_q,$$

where $\overline{X}, \overline{Y}$ denote the images of X, Y in T. It follows that T is a finite ring in which the ring operations can be performed efficiently, in the sense of §3.

Let $Q = (\overline{X} : \overline{Y} : 1) \in E(T)$, and define the endomorphism $\sigma: E(T) \to E(T)$ by the same formula as ϕ above: $\sigma(x : y : z) = (x^q : y^q : z^q)$. As we shall see in a moment, the points Q and $\sigma(Q)$ have order l, and σ satisfies the equation $\sigma^2 - t\sigma + q = 0$ in the endomorphism ring of $E(T)$. Therefore $t \bmod l$ is characterized by the equality

$$\sigma^2(Q) + q \cdot Q = t \cdot \sigma(Q).$$

Thus, to determine $t \bmod l$ one can simply calculate the left-hand side of this equality, and compare it with $0 \cdot \sigma(Q), 1 \cdot \sigma(Q), 2 \cdot \sigma(Q), \ldots$. Here the calculations in $E(T)$ can be done as in §3.

To establish the properties of Q and σ that we used we consider the set V of points $P \in E(K)$ of order l. For each such $P = (x_P : y_P : 1)$ there is a unique $\mathbf{F}_q$-linear ring homomorphism $T \to K$ sending $\overline{X}, \overline{Y}$ to x_P, y_P, respectively. It is straightforward to check that the combined ring homomorphism $T \to \prod_{P \in V} K$ is *injective*, so that $E(T)$ may be considered as a subgroup of $\prod_{P \in V} E(K)$. Since Q corresponds to $(P)_{P \in V}$, it has order l. Also, σ is the restriction to $E(T)$ of the automorphism of $\prod_{P \in V} E(K)$ that on each coordinate is given by ϕ; hence the equality $\sigma^2 - t\sigma + q = 0$ is a consequence of the equality $\phi^2 - t\phi + q = 0$. Clearly, σ is injective, so $\sigma(Q)$ has order l. This concludes our sketch of Schoof's algorithm.

The algorithm is completely deterministic, and it can be shown to run in time $O((\log q)^8)$. (This is slightly better than Schoof [**30**], who has $O((\log q)^9)$.) However, it seems that the algorithm is not suited for practical computations.

We remark that Schoof's algorithm does not calculate the structure of the abelian group $E(\mathbf{F}_q)$. It is known that $E(\mathbf{F}_q) \cong \mathbf{Z}/d_1\mathbf{Z} \times \mathbf{Z}/d_2\mathbf{Z}$ for certain positive integers d_1, d_2 for which d_1 divides d_2, and that d_1 divides

$$\gcd(\#E(\mathbf{F}_q), q - 1).$$

V. Miller has shown that if the prime factorization of the latter gcd is known, one can find d_1 and d_2 by means of a probabilistic algorithm that has expected running time $O((\log q)^{c_2})$ for some $c_2 > 0$. For an account of this algorithm, which depends on the *Weil pairing*, we refer to [**22**].

The fourth method to calculate $\#E(\mathbf{F}_q)$ applies only to curves E that are obtained in a special way. For the sake of simplicity we restrict the discussion to the case that q is a *prime number*.

The *complex multiplication field* of the elliptic curve E over the prime field $\mathbf{F}_q$ is defined to be the field $L = \mathbf{Q}((t^2 - 4q)^{1/2})$, where $t \in \mathbf{Z}$ is such that $\#E(\mathbf{F}_q) = q + 1 - t$. This is an imaginary quadratic field, and its ring of integers

A contains a zero π of the polynomial X^2-tX+q. We have $\pi+\bar{\pi}=t$, $\pi\bar{\pi}=q$, and $\#E(\mathbf{F}_q)=(\pi-1)(\bar{\pi}-1)$. This gives an easy way to calculate $\#E(\mathbf{F}_q)$ provided that L is known, which is the case for certain special curves. We illustrate this by means of two examples that were basically known to Gauss. For proofs, see [**15**, Chapter 18] and also [**12**, §7; **5**].

Let it first be assumed that $q \equiv 1 \bmod 3$ and that the curve $E = E_{a,b}$ has $a = 0$. Then one can prove that $L = \mathbf{Q}(\sqrt{-3})$. The ring of integers A of L is given by $A = \mathbf{Z}[(1+\sqrt{-3})/2]$. To find the element $\pi \in A$ with $\#E(\mathbf{F}_q) = (\pi-1)(\bar{\pi}-1)$ and $\pi\bar{\pi} = q$ one starts by finding an *ideal* $\mathfrak{q}$ with $Aq = \mathfrak{q}\bar{\mathfrak{q}}$, as follows.

One first determines an integer d with $d^2 \equiv -3 \bmod q$. This can be done in one of three ways. The first is to apply general zero-finding routines for polynomials over finite fields, see [**17**, §4.6.2]. The second is to apply a square root extraction algorithm as in [**34**]. The third is to draw elements $u \in \mathbf{F}_q^*$ until one finds one for which $u^{(q-1)/3} \neq 1$ and to put $d \equiv 2u^{(q-1)/3} + 1 \bmod q$. Each of these three methods is probabilistic and practical.

Suppose now that d has been determined. Adding q to d, if necessary, we may assume that d is *odd*. Then $\mathfrak{q} = \mathbf{Z}q + \mathbf{Z}(d+\sqrt{-3})/2$ is a prime ideal of A dividing q, and $\mathfrak{q}\bar{\mathfrak{q}} = Aq$.

Next one determines an element $\pi \in \mathfrak{q}$ for which $\mathfrak{q} = A\pi$. This can be done by searching for the shortest nonzero vector of $\mathfrak{q}$, for which there exist standard reduction algorithms. Alternatively, one can calculate $\gcd(q, (d+\sqrt{-3})/2)$ by means of the Euclidean algorithm, which is valid in A. Notice that π is only uniquely determined by $\mathfrak{q}$ up to units of A, of which there are six.

Now let ζ be the unique sixth root of unity in A for which $b^{(q-1)/6} \equiv \zeta \bmod \mathfrak{q}$; here b is such that $E = E_{0,b}$. Multiplying π by a suitable sixth root of unity we can achieve that $\pi \equiv \bar{\zeta} \bmod 2\sqrt{-3}$. Then one has

$$\#E(\mathbf{F}_q) = (\pi-1)(\bar{\pi}-1) = q + 1 - 2\,\mathrm{Re}(\pi).$$

It can be proved that $E(\mathbf{F}_q)$ is isomorphic to $A/(\pi-1)A$ as an abelian group, so that this method gives the group structure as well.

In the second example that we give we assume that the prime q satisfies $q \equiv 1 \bmod 4$ and that the curve $E = E_{a,b}$ has $b = 0$. Then one can prove that $L = \mathbf{Q}(\mathrm{i})$ with $\mathrm{i}^2 = -1$. It has ring of integers $A = \mathbf{Z}[\mathrm{i}]$. As before, one can find a prime ideal $\mathfrak{q}$ of A such that $\mathfrak{q}\bar{\mathfrak{q}} = Aq$ and an element $\pi \in \mathfrak{q}$ such that $\mathfrak{q} = A\pi$. Denote by ζ the unique fourth root of unity in A for which $(-a)^{(q-1)/4} \equiv \zeta \bmod \mathfrak{q}$. Multiplying π by a suitable fourth root of unity we may assume that $\pi \equiv \bar{\zeta} \bmod 2(1+\mathrm{i})$, and then one has $\#E(\mathbf{F}_q) = (\pi-1)(\bar{\pi}-1)$.

We briefly sketch how these results can be generalized to any imaginary quadratic field L. Let A be the ring of integers of L, and denote by j_L the j-invariant of the elliptic curve $\mathbf{C}/A$ over $\mathbf{C}$ (cf. [**35**, Chapter VI]). It is known that j_L is a zero of an irreducible polynomial $F_L \in \mathbf{Z}[X]$ with leading coefficient 1 and degree equal to the class number of L. Methods to calculate F_L can be found in [**37**]; see also the last section of [**30**]. The cases $j = 0$ and $j = 1728$

correspond to the fields $L = \mathbf{Q}(\sqrt{-3})$ and $\mathbf{Q}(i)$ that we just considered; let these now be excluded.

Let q be a prime number that does not divide the discriminant of L, and suppose that $q > 3$. Then there are methods, analogous to those discussed above, to decide whether there exists $\pi \in A$ with $\pi\bar{\pi} = q$, and to find such an element π if it does exist; it is unique up to conjugation and sign. Suppose that indeed π exists. Then it can be shown that the polynomial $(F_L \bmod q) \in \mathbf{F}_q[X]$ splits into distinct linear factors. Denote by j any zero of this polynomial in $\mathbf{F}_q$. One can prove that $j \neq 0, 1728$. Writing $k = j/(1728 - j) \in \mathbf{F}_q^*$ we now consider the two elliptic curves

$$E = E_{3k,2k}, \qquad E' = E_{3kc^2,2kc^3}$$

over $\mathbf{F}_q$, where $c \in \mathbf{F}_q$ is any nonsquare. Then L is the complex multiplication field of each of the two curves E, E', and the two numbers $\#E(\mathbf{F}_q), \#E'(\mathbf{F}_q)$ are the same as the two numbers $(\pi - 1)(\bar{\pi} - 1), (-\pi - 1)(-\bar{\pi} - 1)$. Presumably there is an easy rule to tell which curve belongs to which number, but I do not know what it is. In practice one can decide between the two cases by picking a point $P \in E(\mathbf{F}_q)$ at random and using that P is annihilated by $\#E(\mathbf{F}_q)$.

This concludes our discussion of the methods to calculate the number of points on an elliptic curve over a finite field.

It is a natural question to ask how the numbers $\#E(\mathbf{F}_q)$ are distributed if q is held fixed and E ranges over all elliptic curves over $\mathbf{F}_q$, up to isomorphism. In particular, one may ask how often a given number occurs as $\#E(\mathbf{F}_q)$. The answer to the latter question, in terms of class numbers of imaginary quadratic orders, is basically due to Deuring [**13**]; see also [**36, 31**]. If q is a *prime* number, then Deuring's result implies that *every* integer of the form $q + 1 - t$ with $|t| < 2\sqrt{q}$ occurs as $\#E(\mathbf{F}_q)$ for some elliptic curve E over $\mathbf{F}_q$. Moreover, it can be deduced that if E is uniformly distributed over all elliptic curves over $\mathbf{F}_q$, then $\#E(\mathbf{F}_q)$ is approximately uniformly distributed over the numbers near $q + 1$. More accurately, one has the following proposition, which is useful for the analysis of some of the algorithms to be presented in §§5 and 6.

PROPOSITION. *There are positive effectively computable constants c_3 and c_4 such that for any prime number $q > 3$ and any set S of integers s for which $|s - (q + 1)| < \sqrt{q}$ one has*

$$\frac{\#S - 2}{2[\sqrt{q}] + 1} \cdot c_3(\log q)^{-1} \leq \frac{N}{q^2} \leq \frac{\#S}{2[\sqrt{q}] + 1} \cdot c_4(\log q) \cdot (\log\log q)^2,$$

where N denotes the number of pairs $(a, b) \in \mathbf{F}_q^2$ that define an elliptic curve $E = E_{a,b}$ over $\mathbf{F}_q$ with $\#E(\mathbf{F}_q) \in S$.

Note that N/q^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 q is in S.

For the proof of the proposition we refer to [**20**, Proposition (1.16)].

5. Primality testing. It was first pointed out in [**5**] and [**8**] that elliptic curves can be used for primality testing. Goldwasser and Kilian [**14**] proved, modulo a reasonable assumption, that this leads to a probabilistic primality testing algorithm of which the expected running time is bounded by a constant power of $\log n$, where n is the number to be tested. The algorithm of Goldwasser and Kilian depends on Schoof's method to count the number of points on an elliptic curve (see §4), and for this reason it is currently not of practical value. Atkin [**2**] developed a variant of this algorithm, in which he employs only the special elliptic curves to which the fourth counting method of §4 applies. His algorithm performs very well in practice, and for the numbers to which it has been applied it beats the method of Adleman et al. [**1**] as implemented by Cohen and A. K. Lenstra [**10**]; these numbers have approximately 200 digits. It seems very hard to give an exact running time estimate of Atkin's algorithm; but a rough heuristic analysis indicates that its expected running time is again bounded by a constant power of $\log n$.

All these methods depend on a result similar to the following theorem, which is the analogue of Theorem 1.

THEOREM 2. *Let n be an integer, $n > 1$, with $\gcd(n,6) = 1$. Let E be an elliptic curve over $\mathbf{Z}/n\mathbf{Z}$, and m, s positive integers with s dividing m. Suppose that there is a point $P \in E(\mathbf{Z}/n\mathbf{Z})$ satisfying*

$$m \cdot P = O,$$

$$\gcd(z_q, n) = 1 \quad \text{for each prime divisor } q \text{ of } s,$$

$$\text{where } m(m/q) \cdot P = (x_q : y_q : z_q).$$

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

The proof, which is analogous to the proof of Theorem 1, is as follows. Let p be a prime divisor of n, and write $Q = (m/s) \cdot P \in E(\mathbf{Z}/n\mathbf{Z})$. Denote by Q_p the image of Q in $E(\mathbf{Z}/p\mathbf{Z})$. From $m \cdot P = O$ it follows that $s \cdot Q = O$, so the order of Q_p divides s. Also, if q is a prime divisor of s, then $s/q \cdot Q_p = (x_q \bmod p : y_q \bmod p : z_q \bmod p)$. This is not the zero point of $E(\mathbf{Z}/p\mathbf{Z})$, since by hypothesis z_q is not divisible by p. Therefore the order of Q_p is not a divisor of s/q, for any prime q dividing s, so this order is equal to s itself. By Lagrange's theorem it follows that $\#E(\mathbf{Z}/p\mathbf{Z})$ is divisible by s. This proves the first assertion of the theorem. If also $s > (n^{1/4}+1)^2$ then Hasse's inequality $(p^{1/2}+1)^2 \geq \#E(\mathbf{Z}/p\mathbf{Z})$ implies that $p > n^{1/2}$, and this can only be true for all primes p dividing n if n is prime. This proves Theorem 2.

The algorithms of Goldwasser-Kilian and Atkin need the above theorem only in the case that s is *prime*, so that only $q = s$ has to be considered in the second hypothesis on P in the above theorem. The following schematic description fits both algorithms.

Let n be a large positive integer that one suspects to be a prime number (cf. the remarks in the introduction). To prove that n is prime one proceeds as follows.

(a) One selects an elliptic curve E over $\mathbf{Z}/n\mathbf{Z}$ and a positive integer m such that the following conditions are satisfied:

(i) $m < (\sqrt{n}+1)^2$, and *if* n is prime *then* $\#E(\mathbf{Z}/n\mathbf{Z}) = m$;

(ii) there are integers $k > 1$ and $q > (n^{1/4}+1)^2$ such that $m = kq$ and such that q is probably prime.

Here *probably prime* means that q passes a pseudoprime test as in [**17**, p. 379], cf. the introduction. To find *one* pair E, m satisfying (i) and (ii), both the algorithm of Goldwasser-Kilian and Atkin's algorithm generate *many* pairs E, m satisfying (i); we shall see below how this is done. It is then hoped that at least one of these pairs satisfies (ii) as well. To check whether a given pair E, m satisfies (ii), one first subjects m to a factoring algorithm that is efficient in finding small factors, such as trial division, or the Pollard $(p-1)$-method (see §2), or the elliptic curve method (see §6); next one lets k be equal to the product of the small prime factors of m that are found, and one puts $q = m/k$; finally, one checks whether $k > 1$ and whether q is probably prime in the sense explained above. (Goldwasser-Kilian require that in fact $k = 2$ in (ii); this makes it even easier to check (ii).)

(b) Now suppose that E, m, k, q as in (a) have been found. Then one picks a random point P of the form $(x_P : y_P : 1)$ in $E(\mathbf{Z}/n\mathbf{Z})$. This is done as in the second counting algorithm explained in §4. (This algorithm works if $\mathbf{Z}/n\mathbf{Z}$ is a field, which one believes to be the case; for the algorithm to work it is not necessary that one has a *proof* that $\mathbf{Z}/n\mathbf{Z}$ is a field!) Next one calculates $Q = k \cdot P$. One now hopes that $Q \neq O$; it can be proved that this is the case for more than half of all choices of P, if n is actually prime. If $Q = O$ one picks another point $P \in E(\mathbf{Z}/n\mathbf{Z})$, and one keeps trying until $Q = k \cdot P \neq O$. Suppose now that $Q \neq O$. Then one checks that $q \cdot Q = O$, as must be the case if n is prime (by $q \cdot Q = m \cdot P$ and (i) above). Finally one checks that $\gcd(z, n) = 1$, if $Q = (x : y : z)$; this must also be the case if n is prime, since $Q \neq O$.

(c) The final stage of the algorithm consists of proving that q is prime. This can be done by a recursive application of the algorithm, or, if q is below a certain bound, by a more direct method. Notice that $q = m/k < (\sqrt{n}+1)^2/2$, so that the depth of the recursion is $O(\log n)$.

If (a), (b), and (c) have been performed successfully, then n is indeed a prime number. This follows from Theorem 2, with $s = q$.

It remains to explain how to find many pairs E, m as in (i). In the Goldwasser-Kilian algorithm this is done as follows. First one draws $a, b \in \mathbf{Z}/n\mathbf{Z}$ at random until $4a^3 + 27b^2 \neq 0$; this happens with probability $(n-1)/n$, if n is indeed prime. Next one checks that $\gcd(n, 4a^3 + 27b^2) = 1$, as should be the case if n is prime. Now one puts $E = E_{a,b}$, and by means of Schoof's algorithm one calculates a number m such that (i) holds. If Schoof's algorithm doesn't work then n is not prime. (If n is not prime, then it is unlikely but not impossible that

Schoof's algorithm calculates a number m; it is an interesting question which information about n this would provide, and what the significance of m would be.)

Atkin's method to find pairs E, m as in (i) is different. Consider the sequence

$$-3, -4, -7, -8, -11, -15, -19, -20, \ldots$$

of discriminants of imaginary quadratic fields; an integer belongs to this sequence if and only if it is negative, not divisible by the square of an odd prime number, and in one of the residue classes 1 mod 4, 8 mod 16, 12 mod 16. For each Δ in a suitable beginning segment of this sequence, one decides whether the ring of integers $A = \mathbf{Z}[(\Delta + \sqrt{\Delta})/2]$ of the imaginary quadratic field $L = \mathbf{Q}(\sqrt{\Delta})$ contains an element π with $n = \pi\bar{\pi}$, and one finds such an element π if it exists; the probabilistic methods to do this that we referred to in §4 are successful provided that n is prime, but, as above, do not require a *proof* that n is prime. The discriminants for which π does not exist are discarded, and the remaining discriminants Δ each give rise to six (if $\Delta = -3$) or four (if $\Delta = -4$) or two (if $\Delta \leq -7$) pairs E, m as in (i), as explained in §4.

For most values of Δ it is easier to determine the values of m than to calculate the coefficients a, b defining E; hence, it is wise to test whether m satisfies (ii) before calculating a, b.

This finishes the description of the primality tests of Goldwasser-Kilian and Atkin.

The running time of a suitable version of the Goldwasser-Kilian algorithm can be analyzed with the help of the proposition stated in §4. The result is expressed in the following two theorems. The first one states that if a certain standard conjecture concerning the distribution of primes is true, then the algorithm runs in expected polynomial time. The second theorem asserts that in any case this is true for almost all input primes n.

THEOREM 3. *Suppose that there are positive constants c_5 and c_6 such that for all real numbers $x \geq 2$ the number of primes p with $x \leq p \leq x + \sqrt{2x}$ is at least $c_5\sqrt{x}(\log x)^{-c_6}$. Then on any prime input n, the Goldwasser-Kilian algorithm proves the primality of n in expected time $O((\log n)^{10+c_6})$.*

For the proof we refer to [**14**]. (The exponent $10 + c_6$ is 1 less than the exponent in [**14**]. This is due to the corresponding improvement in Schoof's algorithm.)

THEOREM 4. *There exist positive constants c_7 and c_8 such that for all integers $k \geq 2$ the fraction of the set of primes n that have k binary digits and for which the expected running time of the Goldwasser-Kilian algorithm is $\leq c_7(\log n)^{11}$ is at least*

$$1 - c_8 2^{-k^{1/\log\log k}}.$$

For the proof we again refer to [**14**]. It employs a theorem of Heath-Brown, which states that the hypothesis made in Theorem 3 is true in a certain average sense.

6. Factorization. We describe a method to factor integers that depends on the use of elliptic curves. It is the analogue of Pollard's $(p-1)$-method described in §2.

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 $\mathbf{Z}/n\mathbf{Z}$ and $P \in E(\mathbf{Z}/n\mathbf{Z})$. This can be done by choosing $a, x, y \in \mathbf{Z}/n\mathbf{Z}$ at random, putting $P = (x : y : 1)$, and letting E be defined by the pair (a, b), where b is chosen such that $P \in E(\mathbf{Z}/n\mathbf{Z})$; so $b = y^2 - x^3 - ax$. To be certain that E is an elliptic curve one should check that $\gcd(4a^3 + 27b^2, n) = 1$. As in Pollard's $(p-1)$-method, one now selects a positive integer k that is divisible by many small prime powers, for example, $k = \operatorname{lcm}\{1, 2, \ldots, w\}$ for a suitable bound w. Next one calculates the point $k \cdot P \in E(\mathbf{Z}/n\mathbf{Z})$. This can be done by $O(\log k)$ duplications and additions in the group $E(\mathbf{Z}/n\mathbf{Z})$. If $k \cdot P = (x : y : z)$, one calculates $\gcd(z, n)$. One stops if this gcd is a nontrivial divisor of n. If, on the other hand, this gcd equals 1 or n, then one changes the pair (E, P) and starts all over again. The latter option is not available in Pollard's method.

As for the Pollard $(p-1)$-method, one can show that a given pair (E, P) is likely to be successful in this algorithm if n has a prime divisor p for which $\#E(\mathbf{Z}/p\mathbf{Z})$ is built up from small primes only. The probability for this to happen increases with the number of pairs (E, P) that one tries.

We refer to [**20**] for the running time analysis of a variant of the elliptic curve factoring algorithm. Using the proposition from §4 and properties of modular curves one finds an upper bound for the expected running time of the algorithm. This upper bound is expressed in terms of the probability that a random number in the interval $(p + 1 - \sqrt{p}, p + 1 + \sqrt{p})$ has all its prime factors below a certain bound, where p denotes the least prime dividing n. To estimate the latter probability we need the following unproved conjecture from analytic number theory.

For a real number $x > e$, define

$$L(x) = e^{\sqrt{\log x \log\log x}}.$$

A theorem of Canfield, Erdös, and Pomerance [**7**, Corollary to Theorem 3.1] implies the following. Let α be a positive real number. Then the probability that a random positive integer $m \leq x$ has all its prime factors $\leq L(x)^{\alpha}$ is $L(x)^{-1/(2\alpha)+o(1)}$, for $x \to \infty$. The conjecture that we need is that the same result is valid if m is a random integer in the interval $(x - \sqrt{x}, x + \sqrt{x})$.

Assuming this conjecture, one arrives at the following running time estimate for the elliptic curve factoring algorithm. Let $n \in \mathbf{Z}$, $n > 1$, be the integer that one wishes to factor, and assume that n is not divisible by 2 or 3 and that it is not a prime power. Let further g be any positive integer. Then the variant of

the elliptic curve factoring algorithm described in [**20**] finds with probability at least $1-e^{-g}$ a nontrivial divisor of n within time $gK(p)(\log n)^2$, where p denotes the smallest prime divisor of n and $K: \mathbf{R}_{>0} \to \mathbf{R}_{>0}$ is a function with

$$K(x) = e^{\sqrt{(2+o(1))\log x \log\log x}} \quad \text{for } x \to \infty.$$

The algorithm may be repeated on the divisors that are found, until the complete prime factorization of n is obtained. The conjectural running time estimate will then also contain terms $gK(p')(\log n)^2$ corresponding to the other prime divisors p' of n, with the exception of the largest one. In all cases one may expect the total factoring time to be at most $L(n)^{1+o(1)}$ for $n \to \infty$, with L as above. The worst case occurs if the second largest prime divisor of n is not much smaller than $\sqrt{n}$, so that n is the product of some small primes and two large primes that are of the same order of magnitude.

Several other factoring methods have been proposed for which, conjecturally, the running time is $L(n)^{1+o(1)}$ for $n \to \infty$, such as the class group method [**29**] and the quadratic sieve [**26**]; see also the discussion in [**11**]. However, for these other methods the running time is basically independent of the size of the prime factors of n, whereas the elliptic curve method is substantially faster if the smallest prime factor of n is much smaller than $\sqrt{n}$.

The storage requirement of the elliptic curve factoring method is only $O(\log n)$. This is also true for the class group method [**29**], but all other known factoring algorithms of conjectured speed $L(n)^{1+o(1)}$ have a storage requirement that is a positive power of $L(n)$.

We refer to [**23**, **6**] for modifications of the elliptic curve method that improve its practical performance. It turns out that, with these modifications, the elliptic curve method is one of the fastest integer factorization methods that is currently used in practice. The quadratic sieve algorithm still seems to perform better on integers that are built up from two prime numbers of the same order of magnitude; such integers are of interest in cryptography [**28**].

ACKNOWLEDGMENT. Part of the work on this lecture was done at the University of Chicago. I thank this institution for its hospitality and support.

REFERENCES

1. L. M. Adleman, C. Pomerance, and R. S. Rumely, *On distinguishing prime numbers from composite numbers*, Ann. of Math. (2) **117** (1983), 173–206.

2. A. O. L. Atkin, in preparation.

3. E. Bach and J. Shallit, *Factoring with cyclotomic polynomials* (26th Annual Sympos. Foundations of Comput. Sci. (FOCS), Portland, 1985) IEEE Comput. Soc. Press, Washington, 1985, pp. 443–450.

4. H. Bass, *Algebraic K-theory*, Benjamin, New York, 1968.

5. W. Bosma, *Primality testing using elliptic curves*, Report 85-12, Math. Inst. Universiteit van Amsterdam, 1985.

6. R. P. Brent, *Some integer factorization algorithms using elliptic curves*, Research report CMA-R32-85, The Australian National Univ., Canberra, 1985.

7. E. R. Canfield, P. Erdös, and C. Pomerance, *On a problem of Oppenheim concerning "Factorisatio Numerorum"*, J. Number Theory **17** (1983), 1–28.

8. D. V. Chudnovsky and 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 Res. Center, Yorktown Heights, N.Y., 1985.

9. H. Cohen and H. W. Lenstra, Jr., *Primality testing and Jacobi sums*, Math. Comp. **42** (1984), 297–330.

10. H. Cohen and A. K. Lenstra, *Implementation of a new primality test*, Math. Comp. **48** (1987), 103–121 and S1–S4.

11. D. Coppersmith, A. M. Odlyzko, and R. Schroeppel, *Discrete logarithms in* GF(p), Algorithmica **1** (1986), 1–15.

12. H. Davenport and H. Hasse, *Die Nullstellen der Kongruenzzetafunktionen in gewissen zyklischen Fällen*, J. Reine Angew. Math. **172** (1934), 151–182.

13. M. Deuring, *Die Typen der Multiplikatorenringe elliptischer Funktionenkörper*, Abh. Math. Sem. Hansischen Univ. **14** (1941), 197–272.

14. S. Goldwasser and J. Kilian, *Almost all primes can be quickly certified* (Proc. 18th Annual ACM Sympos. on Theory of Computing (STOC, Berkeley, 1986), The Association for Computing Machinery, New York, 1986, pp. 316–329.

15. K. Ireland and M. Rosen, *A classical introduction to modern number theory*, Graduate Texts in Math., vol. 84, Springer-Verlag, New York, 1982.

16. N. M. Katz and B. Mazur, *Arithmetic moduli of elliptic curves*, Princeton Univ. Press, Princeton, N.J., 1985.

17. D. E. Knuth, *The art of computer programming*, vol. 2: *Seminumerical algorithms*, 2nd ed., Addison-Wesley, Reading, Mass., 1981.

18. S. Lang and H. Trotter, *Frobenius distributions in* GL_2*-extensions*, Lecture Notes in Math., vol. 504, Springer-Verlag, Berlin, 1976.

19. H. Lange and W. Ruppert, *Complete systems of addition laws on abelian varieties*, Invent. Math. **79** (1985), 603–610.

20. H. W. Lenstra, Jr., *Factoring integers with elliptic curves*, Ann. of Math. (to appear).

21. H. W. Lenstra, Jr. and R. Tijdeman (Editors), *Computational methods in number theory*, Math. Centre Tracts, no. 154/155, Mathematisch Centrum, Amsterdam, 1982.

22. V. S. Miller, *Short programs for functions on curves*, IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1986.

23. P. L. Montgomery, *Speeding the Pollard and elliptic curve methods of factorization*, Math. Comp. **48** (1987), 243–264.

24. H. C. Pocklington, *The determination of the prime and composite nature of large numbers by Fermat's theorem*, Proc. Cambridge Philos. Soc. **18** (1914–16), 29–30.

25. J. M. Pollard, *Theorems on factorization and primality testing*, Proc. Cambridge Philos. Soc. **76** (1974), 521–528.

26. C. Pomerance, *Analysis and comparison of some integer factoring algorithms*, in [**21**], pp. 89–139.

27. H. Riesel, *Prime numbers and computer methods for factorization*, Progr. Math., vol. 57, Birkhäuser, Boston, 1985.

28. R. L. Rivest, A. Shamir, and L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Comm. ACM **21** (1978), 120–126.

29. C. P. Schnorr and H. W. Lenstra, Jr., *A Monte Carlo factoring algorithm with linear storage*, Math. Comp. **43** (1984), 289–311.

30. R. J. Schoof, *Elliptic curves over finite fields and the computation of square roots mod p*, Math. Comp. **44** (1985), 483–494.

31. R. J. Schoof, *Nonsingular plane cubic curves over finite fields*, J. Combin. Theory Ser. B (to appear).

32. J. L. Selfridge and M. C. Wunderlich, *An efficient algorithm for testing large numbers for primality*, Proc. Fourth Manitoba Conf. Numerical Math., University of Manitoba, Congressus Numerantium XII, Utilitas Math., Winnipeg, 1975, pp. 109–120.

33. D. Shanks, *Class number, a theory of factorization, and genera*, Proc. Sympos. Pure Math. vol. 20, Amer. Math. Soc., Providence, R.I., 1971, pp. 415–440.

34. ——, *Five number-theoretic algorithms*, Proc. Second Manitoba Conf. Numerical Math., University of Manitoba, Congressus Numerantium VII, Utilitas Math., Winnipeg, 1973, pp. 51–70.

35. J. H. Silverman, *The arithmetic of elliptic curves*, Graduate Texts in Math., vol. 106, Springer-Verlag, New York, 1986.

36. W. C. Waterhouse, *Abelian varieties over finite fields*, Ann. Sci. École Norm. Sup. (4) **2** (1969), 521–560.

37. H. Weber, *Lehrbuch der Algebra*, vol. III, Friedrich Vieweg und Sohn, Braunschweig, 1908; reprinted by Chelsea Publishing Company, New York.

38. H. C. Williams, *Primality testing on a computer*, Ars Combin. **5** (1978), 127–185.

39. ——, *A $p+1$ method of factoring*, Math. Comp. **39** (1982), 225–234.

MATHEMATISCH INSTITUUT, UNIVERSITEIT VAN AMSTERDAM, 1018 WB AMSTERDAM, THE NETHERLANDS