Elliptic Curves in Algorithmic Number Theory and Cryptography

Otto Forster

§1 Applications in Algorithmic Number Theory

In this section we describe briefly the use of elliptic curves over finite fields for two fundamental problems in algorithmic number theory, namely factorization and proving primality of large integers.

1.1 Factorization. The elliptic curve factorization method of H. Lenstra is a generalization of the so-called $(p-1)$-factorization algorithm of Pollard. The common setup for both methods is the following: Suppose we want to find a factor of some large integer N. Let there be given a functor that associates to N a group $G(N)$ and to any prime divisor $p \mid N$ a group $G(p)$ and a group homomorphism $\beta_p : G(N) \to G(p)$ with the following property: If $x \in G(N) \smallsetminus \{e\}$ is a nontrivial element lying in the kernel of one of the β_p (for an unknown prime divisor $p \mid N$), then a nontrivial divisor of N can be easily calculated. In the case of Pollard's $(p-1)$-method one sets $G(m) := (\mathbb{Z}/m)^*$ for all integers $m > 0$. If an element $\bar{x} = x \bmod N \in (\mathbb{Z}/N)^*$ is in the kernel of the natural homomorphism $\beta_p : (\mathbb{Z}/N)^* \to (\mathbb{Z}/p)^*$ for some prime divisor $p \mid N$ and if $x \not\equiv 1 \bmod N$, then

$$d := \gcd(x-1, N)$$

is a nontrivial divisor of N. But how can we find a nontrivial element in the kernel of β_p if p is unknown? This is possible provided that the order of $G(p)$ is a "smooth" number, i.e. if

$$\#G(p) = q_1^{k_1} q_2^{k_2} \cdot \ldots \cdot q_r^{k_r}$$

with small prime powers $q_i^{k_i}$, say $q_i^{k_i} \le B$ for all i and a given (relatively small) bound B. One then calculates the number

$$Q(B) = \prod_{q \le B} q^{\alpha(q,B)},$$

where $\alpha(q,B) := \max\{k \in \mathbb{N} : q^k \le B\}$. By the prime number theorem, $Q(B)$ has order of magnitude $\exp(B)$. Since by assumption $\#G(p) \mid Q(B)$, for every element

August 2002

This is a preliminary version of an appendix to the forthcoming 2nd edition of the book D. Husemöller: Elliptic Curves, Springer-Verlag

$\xi \in G(p)$ we have $\xi^{Q(B)} = e$. Therefore, if we calculate $y := x^{Q(\mathbb{B})}$ for an arbitrary element $x \in G(N)$, then $y \in \ker(\beta_p)$, because $\beta_p(y) = \beta_p(x)^{Q(B)} = e$. If $y \neq e$, then by assumption a divisor of N can be calculated. Pollard's method is efficient if there is a prime divisor $p \mid N$ such that $p-1$ is a smooth number. But this is not always the case. It was Lenstra's idea to replace the multiplicative group $\mathbb{F}_p^*$ in Pollard's method by an elliptic curve $G(p) = E_{a,b}(\mathbb{F}_p)$. By varying the parameters a, b of the elliptic curve, there is a better chance that the order $\#E_{a,b}(\mathbb{F}_p)$ is a sufficiently smooth number.

Lenstra's algorithm works as follows: To start, we choose random elements $a \in \mathbb{Z}/N$, $P_0 = (x_0, y_0) \in (\mathbb{Z}/N)^2$ and determine a value $b \in \mathbb{Z}/N$ such that

$$y_0^2 \equiv x_0^3 + ax_0 + b \mod N.$$

In rare cases we will have $\gcd(4a^3 + 27b^2, N) \neq 1$. Then we have either found a nontrivial divisor of N and can stop the algorithm or else $N \mid 4a^3 + 27b^2$ and we must start again with new random values a, x_0, y_0.

If $\gcd(4a^3 + 27b^2, N) = 1$, consider the equation

$$Y^2 = X^3 + aX + b.$$

For every prime divisor $p \mid N$, we define the group $G(p) := E_{a,b}(\mathbb{Z}/p)$ as the elliptic curve defined by this equation taken modulo p and set $G(N) := \prod_{p|N} E_{a,b}(\mathbb{Z}/p)$. The homomorphisms $\beta_p : G(N) \to G(p)$ are the natural projections. If we denote by $G(p)' := G(p) \smallsetminus \{O\}$ the affine part of $G(p)$, then $G(N)' := \prod_{p|N} G(p)'$ is the complement of $\bigcup_{p|N} \ker(\beta_p)$. The points of $G(N)'$ can be represented by pairs (x, y) of integers satisfying our equation modulo N. We have already constructed a point $P_0 = (x_0, y_0)$ of $G(N)'$. By the general principle of the factorization algorithm explained above, we must now calculate the multiple $Q(B) \cdot P_0$ (for some suitable choice of B). This can be done in $O(\log Q(B))$ steps by repeated doubling and adding. The group law to add two points $P_1 + P_2 =: P_3$ is given by the formulas

$$x_3 := \lambda^2 - x_1 - x_2, \qquad y_3 := \lambda(x_1 - x_3) - y_1,$$

where the "slope" λ is defined by

$$\lambda := \frac{y_2 - y_1}{x_2 - x_1} \text{ if } x_1 \neq x_2 \quad \text{and} \quad \lambda := \frac{3x_1^2 + a}{2y_1} \text{ if } P_1 = P_2.$$

The only problem in doing these operations in $\mathbb{Z}/N$ is the calculation of the inverses of the denominators. These inverses, if they exist in $\mathbb{Z}/N$, can be calculated by using the extended Euclidean algorithm to calculate the gcd of the denominator and N. If the gcd equals 1, the inverse can be calculated and we can go on. The exceptional case is that the gcd is a number $d \neq 1$. If $d \neq N$, we are in a lucky case because we have found a divisor of N. If one of the elliptic curves $G(p)$ has an order dividing $Q(B)$, an exceptional case must necessarily occur during the calculation of $Q(B) \cdot P_0$, because then $Q(B) \cdot P_0$ cannot be an element of $G(N)'$. If we do not encounter a lucky case we are not completely lost, because we can start again with new random parameters a, x_0, y_0, i.e. with new elliptic curves $G(p)$ with different orders. A nice feature of the elliptic curve factorization algorithm is that it is easily parallelizable, because we can let many computers work on the factorization of the same number N using different elliptic curves.

1.2 Deterministic Primality Tests. There are some very efficient probabilistic primality tests for large integers. An example is the Solovay-Strassen test. This test works as follows. Let N be a large odd integer to be tested for primality. Choose a random integer a with $1 < a < N$ and check whether (1) $\gcd(a, N) = 1$, and (2) $a^{(N-1)/2} \equiv (\frac{a}{N}) \bmod N$, where $(\frac{a}{N})$ is the Jacobi symbol. Of course, if N is prime, these conditions are satisfied (condition (2) is a theorem of Euler). Hence if one of the conditions fails, we are certain that N is not prime. If both conditions hold, we can assert the primality of N only with a certain error probability. Indeed one can show that for composite N condition (2) is satisfied for less than $N/2$ values of a. Hence the error probability is less than $\frac{1}{2}$. (For most N the error probabilty is much less.) By repeating the test m times with independent random values a, the error probability will be less than 2^{-m}. An integer N which has passed successfully sufficiently often a probabilistic primility test is called a "probable prime". For all practical purposes we may assume that N is prime, but this is not a mathematical certainty.

If the prime decomposition of $N-1$ is known, there is a simple deterministic primality test: N is prime if and only if there exists an integer a such that $a^{N-1} \equiv 1 \bmod N$ and $a^{(N-1)/q} \not\equiv 1$ for all prime divisors $q \mid N-1$. An a with this property is then a primitive root modulo N. If N is prime then there exist $\varphi(N-1)$ primitive roots, hence by trying out some random numbers one can be found. But in general $N-1$ (which is the order of $(\mathbb{Z}/N)^*$ in case N is prime) is difficult to factorize. As in the case of Lenstra's factorization method one can try to replace the group $(\mathbb{Z}/N)^*$ by an elliptic curve $E_{a,b}$. By varying the coefficients a, b, the orders of the elliptic curves vary and there is a better chance that at least one of these orders can be factorized. This was the idea of Goldwasser/Kilian. Their primality test is based on the following proposition.

Proposition. *Let N be a probable prime with* $\gcd(6, N) = 1$ *and let a, b be integers with* $\gcd(4a^3 + 27b^2, N) = 1$*. Consider the elliptic curve with affine equation*

$$E = E_{a,b} : \quad Y^2 = X^3 + aX + b.$$

Suppose there exists a prime $q > (\sqrt[4]{N} + 1)^2$ and an affine point $P = (x, y)$ on $E(\mathbb{Z}/N)$ such that $q \cdot P = O$. Then N is prime.

Remark. As in 1.1 we define $E(\mathbb{Z}/N) = \prod_{p|N} E(\mathbb{Z}/p)$. All calculations are done in $\mathbb{Z}/N$. In contrast to 1.1, here an exceptional case where we encounter a denominator, which is a nonzero noninvertible element of $\mathbb{Z}/N$, will rarely occur in practice, because N is a probable prime.

Proof. Assume that N is not prime. Then there exists a prime divisor $p \mid N$ with $p \le \sqrt{N}$. The natural homomorphism

$$E(\mathbb{Z}/N) \longrightarrow E(\mathbb{Z}/p)$$

maps P to a point $\overline{P} = (\bar{x}, \bar{y}) \in E(\mathbb{Z}/p)$ of order q. By the theorem of Hasse (Chap. 13.1, Theorem (1.2)), the order of $E(\mathbb{Z}/p)$ satisfies

$$\#E(\mathbb{Z}/p) < p + 1 + 2\sqrt{p} \le \sqrt{N} + 1 + 2\sqrt[4]{N} = (\sqrt[4]{N} + 1)^2.$$

Therefore it would follow that $q > \#E(\mathbb{Z}/p)$, a contradiction!

The primality test of Goldwasser/Kilian uses this proposition in the following way: Choose random numbers a, b and determine the order $m := \#E_{a,b}(\mathbb{Z}/N)$ by Schoof's algorithm (cf. 2.6), assuming N is prime. By trial division of m by small primes write m as $m = f \cdot u$, where f is the factored and u the unfactored part. If $f \ge 2$ and $u > (\sqrt[4]{N} + 1)^2$, test whether $q := u$ is a probable prime. If this is not the case or if u is not of the required size, start again with new random values a, b. If q is a probable prime, it is in general easy to find a point $P = (x, y)$ on $E_{a,b}(\mathbb{Z}/N)$ of order q. Then by the proposition N is prime provided q is prime. Since $q \le \frac{1}{2}\,\#E_{a,b}(\mathbb{Z}/N)$, this can be tested recursively by the same method. The primality test of Goldwasser/Kilian has expected polynomial running time (polynomial in the number of bits of N), but still is too slow in practice.

Atkin/Morain have devised an improvement which makes this primality test efficient in practice. Instead of choosing random elliptic curves and calculating their order, they construct, using a complex multiplication method, elliptic curves whose order is known a priori. Let $-D$ be the discriminant of an imaginary quadratic number field. If N is prime and the equation $4N = t^2 + Ds^2$ has an integer solution (t, s), then there exists an elliptic curve E over the field $\mathbb{Z}/N$, whose endomorphism ring is the ring of algebraic integers in $\mathbb{Q}(\sqrt{-D})$, and which has $m = \#E(\mathbb{Z}/N) = N + 1 \pm t$ elements. As above, one can test whether m can be written as $m = f \cdot q$, where q is a probable prime with $m/2 \ge q > (\sqrt[4]{N} + 1)^2$. There exists an effective algorithm of Cornacchia to decide whether the Diophantic equation $4N = t^2 + Ds^2$ is solvable and to find a solution in case of existence (of course $(\frac{-D}{N}) = 1$ is a necessary condition). The equation of the elliptic curve E can be constructed in the following way: We first calculate the j-invariant $j_D := j\left(\frac{-D + i\sqrt{D}}{2}\right) \in \mathbb{C}$ with sufficiently high precision. This is an algebraic integer of degree equal to the class number h of the field $\mathbb{Q}(\sqrt{-D})$. Its conjugates are $j(\tau_\nu), \nu = 2, \ldots, h$, where the lattices $\mathbb{Z} + \mathbb{Z}\tau_\nu$ represent the non-principal ideal classes of $\mathbb{Q}(\sqrt{-D})$. By calculating also these conjugates of j_D, we get its minimal polynomial $H_D(T) \in \mathbb{Z}[T]$. This polynomial, taken modulo N, has at least one zero $j_0 \in \mathbb{Z}/N$, which is the j-invariant of the elliptic curve $E(\mathbb{Z}/N)$. From this we can calculate the equation of the elliptic curve. Up to isomorphism, there are only two possibilities, except for $D = -3$ with 6, and $D = -4$ with 4 isomorphism classes.

Incorporating further improvements, the primality test of Atkin/Morain is very efficient and has been used to prove the primality of numbers with more than 1000 decimal digits.

§2 Elliptic Curves in Cryptography

The use of elliptic curves in cryptography is based on the discrete logarithm problem. First we describe this problem in a general group.

2.1 The Discrete Logarithm. Let G be a finite abelian group (we will write it multiplicatively) and let $g \in G$ be a fixed element of known order q. Let $G_0 = \langle g \rangle$ the cyclic subgroup of G generated by g. Then we have an isomorphism of groups

$$\exp_g : \mathbb{Z}/q\mathbb{Z} \longrightarrow G_0, \quad k \mapsto g^k.$$

The inverse map of $\exp_g$ is called the *discrete logarithm* (with respect to basis g)

$$\log_g : G_0 \longrightarrow \mathbb{Z}/q\mathbb{Z}.$$

More concretely, given an element $x \in G_0 = \langle g \rangle$, the discrete logarithm of x is the unique number k mod q such that $x = g^k$.

Popular choices for the group G are the multiplicative group of a finite field or an elliptic curve over a finite field.

The crucial point for the cryptographical applications is that the exponential map can be effectively calculated, whereas the calculation of the logarithm is in general much more complicated. To give an idea of the orders of magnitude involved, the bitsize of the number q (which should be a prime for reasons that we will explain later) is typically between 160 and 1024 (i.e. $q \approx 2^{160}$ up to $q \approx 2^{1024}$). The power g^k can be calculated by the repeated squaring algorithm: If

$$k = \sum_{i=0}^{r} b_i 2^r, \quad b_i \in \{0, 1\}$$

then

$$g^k = \prod_{b_i \neq 0} g^{2^i}$$

and g^{2^i} requires i multiplications. Hence the complexity grows linearly with the number of digits of q. The complexity of the discrete logarithm depends of course on the particular group G. We will discuss this problem later, but we say at this point only that for general elliptic curves the best known algorithms have a complexity growing exponentially with the number of digits of q.

We will now describe two cryptographical applications of the discrete logarithm in the context of a general group.

2.2 Diffie-Hellman Key Exchange. Suppose that two parties, say Alice and Bob, want to take up a confidential communication over an unsecured channel like the

Internet. For this purpose they send their messages encrypted with a secret key that is known only to Alice and Bob. But how can they agree on a common secret key if this information must also be exchanged over the unsecured channel? This can be done by a public key system invented by W. Diffie and M.E. Hellman. First Alice and Bob agree on a triple (G, g, q) consisting of a group G and an element $g \in G$ of order q as in (2.1). It is supposed that the discrete logarithm problem in G is intractable. This (G, g, q) is a public key that need not to be kept secret. For every particular session a new secret key is established in the following manner:

1. Alice chooses a random number $\alpha \in \mathbb{Z}/q\mathbb{Z}$, calculates $a := g^\alpha \in G$ and sends a to Bob. The number α must be kept secret, but a may be known to an adversary.

2. Bob chooses a random number $\beta \in \mathbb{Z}/q\mathbb{Z}$, calculates $b := g^\beta \in G$ and sends b to Alice. Again β must be kept secret.

3. Alice calculates $k_a := b^\alpha \in G$, and Bob calculates $k_b := a^\beta \in G$. Of course

$$k_a = g^{\alpha\beta} = k_b;$$

so they can use $k_a = k_b$ as their common secret key. An adversary knows $a = g^\alpha$ and $b = g^\beta$. To calculate $g^{\alpha\beta}$ from g^α and g^β is known as the Diffie-Hellman problem. For this no better method is known than to calculate α or β by solving the discrete logarithm problem for one of the equations $a = g^\alpha$ or $b = g^\beta$. But this was supposed to be practically impossible.

2.3 Digital Signatures. An electronic document can be easily copied and the copy is completely identical to the original. Therefore, at first sight, it seems that a digital signature can be forged even more easily than can handwritten signature. Therefore it is surprising that a secure digital signature scheme can be established using public key cryptography. The idea is to use signatures that depend on the signed document and that can only be produced using a private (secret) key, whereas verification of the signature is possible using the public key corresponding to the secret key.

There are several digital signature schemes; we will describe one that is a variant of a scheme invented by T. ElGamal. This scheme uses the discrete logarithm and can be formulated for an arbitrary finite abelian group (for example, an elliptic curve over a finite field).

So let (G, g, q) be (as above) a triple where G is a group and $g \in G$ an element of known prime order q and suppose that the discrete logarithm problem in G is intractable. Furthermore let there be given a map $\varphi : G \to \mathbb{Z}/q\mathbb{Z}$. (For example, if G is an elliptic curve over a prime field $\mathbb{F}_p$, for a point $A \in G, A \neq O$, we could define $\varphi(A) = x(A) \bmod q$, where $x(A) \in \{0, 1, 2, \dots, p-1\}$ is the x-coordinate of A.)

1. To set up a public/private key pair for digital signatures, Alice chooses a random number $\xi \in (\mathbb{Z}/q)^*$ and calculates

$$h := g^\xi \in G.$$

The public key is then (G, q, φ, g, h), whereas ξ serves as Alice's private key and must be kept secret. (An adversary can calculate ξ from the public data, provided he can solve the discrete logarithm problem in G, which we supposed to be practically impossible.)

2. To sign a particular message $m \in (\mathbb{Z}/q\mathbb{Z})^*$ (in practice m will be a so called *message digest* or *cryptographic check sum* of a longer document), Alice chooses a new random number $\alpha \in (\mathbb{Z}/q\mathbb{Z})^*$ and calculates

$$a := g^\alpha \in G,$$

and, using her private key ξ,

$$m' := m + \xi\varphi(a) \in \mathbb{Z}/q\mathbb{Z}.$$

If $m' = 0$ (a case which in practice will never occur, because its probability is only $1/q$), another random number α has to be chosen. Then Alice calculates

$$\beta := \alpha^{-1}m' \in (\mathbb{Z}/q\mathbb{Z})^*.$$

The signature of m is

$$\sigma := (a, \beta) \in G \times \mathbb{Z}/q\mathbb{Z},$$

and the signed message is the pair (m, σ).

3. If Bob wants to verify that (m, σ) was indeed signed by Alice, he does the following calculations (which use only the public key)

$$\gamma := m\beta^{-1} \in \mathbb{Z}/q\mathbb{Z}, \qquad \delta := \varphi(a)\beta^{-1} \in \mathbb{Z}/q\mathbb{Z},$$

and

$$c := g^\gamma h^\delta \in G.$$

He accepts the signature if $c = a$. If the message m was properly signed, this is indeed the case, because

$$\begin{aligned} g^\gamma h^\delta &= g^{m\beta^{-1}} g^{\xi\varphi(a)\beta^{-1}} \\ &= g^{(m+\xi\varphi(a))\beta^{-1}} = g^{m'\beta^{-1}} = g^\alpha = a. \end{aligned}$$

2.4 Algorithms for the Discrete Logarithm. Let G be a cyclic group of order q with generator g and $x \in G$. We wish to determine a number $\xi \in \mathbb{Z}/q\mathbb{Z}$ such that

$$x = g^\xi.$$

If q is not prime, but a composite with prime factorization

$$q = \prod p_j^{r_j},$$

it is easy to see that the problem can be reduced to cyclic groups of order p_j. Therefore the discrete logarithm problem is hardest if q is prime.

The baby step/giant step (BSGS) algorithm of Shanks proceeds in the following way: Let $k := \lceil\sqrt{q}\rceil$ be the smallest integer $\geq \sqrt{q}$. The (unknown) discrete logarithm ξ can be written as

$$\xi = nk + m, \quad 0 \leq n, m < k.$$

The equation $x = g^{\xi}$ is equivalent to

$$xg^{-m} = g^{kn}.$$

First, the "giant steps"

$$g^{k\nu}, \quad \nu = 0, 1, \ldots, k-1$$

are calculated and stored in a hash table. Then the "baby steps"

$$xg^{-\mu}, \quad \mu = 0, 1, 2, \ldots$$

are calculated one after the other and compared with the stored values until a collision

$$xg^{-m} = g^{kn}$$

is found. The discrete logarithm is then $\xi = (kn + m) \bmod q$. If efficient hashing techniques are used for storing and searching, this algorithm requires roughly $O(\sqrt{q})$ steps. The memory requirement (for the giant steps) is also $O(\sqrt{q})$. However there exist probabilistic variants (Pollard's rho and lambda method) which use only a small constant amount of memory and have the same time complexity $O(\sqrt{q})$.

Remark. The complexity $O(\sqrt{q})$ is an *exponential* complexity considering it (as customary) as a function of the number of binary digits of q.

To be safe against this algorithm (i.e. to make the discrete logarithm problem intractable), q should be by today's (2002) standards at least 2^{160}. The number of required steps would then be $> 2^{80} \approx 1.2 \cdot 10^{24}$.

For special groups there exist more efficient algorithms for the discrete logarithm. For example, for the multiplicative group $\mathbb{F}_q^*$ of a finite field there exist *subexponential* algorithms (index calculus method, number field sieve). Subexponential complexity is between polynomial and exponential complexity.

For general elliptic curves over finite fields no better algorithms for the discrete logarithm problem are known than the general purpose $O(\sqrt{q})$ algorithms. However, for elliptic curves with special properties, one can do better. For example, let E be a supersingular elliptic curve over $\mathbb{F}_p$, so that $E(\mathbb{F}_p)$ has $n = p + 1$ elements. Using the Weil pairing

$$E[n] \times E[n] \longrightarrow \mu_n$$

and the fact that $\mu_n = \mu_{p+1}$ is a subgroup of $\mathbb{F}_{p^2}^*$, one can embed $E(\mathbb{F}_p)$ into the multiplicative group $\mathbb{F}_{p^2}^*$ and use the more efficient algorithms in $\mathbb{F}_{p^2}^*$ to solve the discrete logarithm problem. For several other special classes of elliptic curves algorithms with complexity better than $O(\sqrt{q})$ are known. So the recommendation for the application of elliptic curves in cryptography is to use "random" elliptic curves (i.e. curves with random coefficients) in the hope that the special algorithms for the discrete logarithm that have been found or may be found in the future do not apply to them. As we have seen, to make the discrete logarithm problem difficult, the order of the group should be a prime number or have at least a large prime factor. So the problem arises of counting the number of points of the randomly chosen elliptic curves. If one has efficient algorithms for this purpose, one chooses random elliptic curves and determines their order. If the order is not satisfactory, the curve is thrown away and a new random curve is chosen, until a good one is found.

2.5 Counting the Number of Points. A straightforward way to determine the number of points of an elliptic curve E over the prime field $\mathbb{F}_p$, (p an odd prime), given by the equation

$$Y^2 = X^3 + aX + b = P_3(X)$$

is to use the Legendre symbol. For a given $x \in \mathbb{F}_p$, the equation $Y^2 = P_3(x)$ has 2, 1 or 0 solutions in $\mathbb{F}_p$ if $\left(\frac{P_3(x)}{p}\right)$ equals $+1$, 0 or -1 respectively. Therefore, taking into account also the point at infinity, it follows

$$\#E(\mathbb{F}_p) = 1 + \sum_{x \in \mathbb{F}_p} \left\{ 1 + \left(\frac{P_3(x)}{p}\right) \right\} = (p+1) + \sum_{x \in \mathbb{F}_p} \left(\frac{P_3(x)}{p}\right).$$

However, this method has complexity $O(p)$ and can be used only for small primes p (say up to 10^6).

A better method with complexity $O(\sqrt[4]{q})$ is an adaption of Shanks's baby step/giant step algorithm. Let E be an elliptic curve over a finite field $\mathbb{F}_q$. By the theorem of Hasse, the order of E lies in the "Hasse interval"

$$H := \{n \in \mathbb{N} : |n - (q+1)| \le 2\sqrt{q}\}.$$

One chooses a random point $P \in E(\mathbb{F}_q)$ and determines by the BSGS algorithm an integer $N \in H$ such that $N \cdot P = O$. Since H has $1 + 2\lfloor 2\sqrt{q} \rfloor$ elements, this can be done with about $2\sqrt[4]{q}$ giant and baby steps. If N is the only element of the Hasse intervall with $N \cdot P = O$, this is the order of $E(\mathbb{F}_q)$. For orders up to 10^{24}, this method is effective in practice. But the elliptic curves used in cryptography are still larger, so other methods are needed.

2.6 Schoof's Algorithm. Recall that for an elliptic curve E defined over a finite field $\mathbb{F}_q$ the Frobenius automorphism $\phi = \phi_q : E \to E$ satisfies a quadratic equation

$$\phi^2 - c\phi + q = 0,$$

where the trace c is connected to the order N of the elliptic curve by

$$N = \#E(\mathbb{F}_q) = q + 1 - c.$$

The idea of Schoof is to calculate $c_\ell := c \bmod \ell$ for various small primes ℓ by restricting the Frobenius automorphism to the group of ℓ-division points $E[\ell] \subset E$, which is invariant under ϕ. If the characteristic p of the field $\mathbb{F}_q$ is bigger than ℓ, then

$$E[\ell] \cong \mathbb{Z}/\ell\mathbb{Z} \times \mathbb{Z}/\ell\mathbb{Z} = \mathbb{F}_\ell^2$$

is a 2-dimensional vector space over $\mathbb{F}_\ell$ and the restriction $\phi \mid E[\ell]$, which we denote again by ϕ, satisfies the characteristic equation

$$\phi^2 - c_\ell \phi + q_\ell = 0,$$

with $q_\ell = q \bmod \ell$. The trace c_ℓ can be calculated by choosing a point $P \in E[\ell] \smallsetminus \{0\}$ and solving the equation

$$\phi^2(P) + q_\ell P = c_\ell \phi(P).$$

If c_ℓ is known for all $\ell \in \{\ell_1, \ldots, \ell_r\}$, then by the Chinese remainder theorem we can calculate c modulo $L := \prod \ell_\nu$. If L is greater than the length $4\sqrt{q}$ of the Hasse interval, c and therefore $N = \#E(\mathbb{F}_q)$ is uniquely determined. Even if $L < 4\sqrt{q}$, then there are at most $\lceil 4\sqrt{q}/L \rceil$ possible values for N. Using an appropriate BSGS method, one can then determine the correct value of N in about $\sqrt{4\sqrt{q}/L}$ steps.

How can we find a point $P \in E[\ell] \smallsetminus \{0\}$? For odd ℓ, the x-coordinates of these points are the roots of the ℓ-division polynomial $\Psi_\ell(T) \in \mathbb{F}_q[T]$, which is a polynomial of degree $(\ell^2 - 1)/2$ (because the $\ell^2 - 1$ points of $E[\ell] \smallsetminus \{0\}$ come in pairs $\pm P$ having the same x-coordinate), cf. [5], Chap. 13.9. Using the recursion formulas, the division polynomials can be easily calculated. In general, Ψ_ℓ neither has a zero in the ground field $\mathbb{F}_q$ nor is it irreducible. Suppose we know an irreducible factor $F(T)$ of degree r of the polynomial $\Psi_\ell(T)$. Then the field $K := \mathbb{F}_q[T]/(F(T))$ is isomorphic to $\mathbb{F}_{q^r}$ and the element $t := T \bmod F(T) \in K$ is the x-coordinate of an ℓ-division point. If the element $P_3(t)$ is the square of an element $s \in K$, then $(t, s) \in K^2$ is an ℓ-division point of the elliptic curve, otherwise one has to pass to a quadratic extension of K. To avoid the case distinction it is convenient, instead of working with the curve

$$E: \quad Y^2 = X^3 + aX + b =: P_3(X),$$

to work with the twisted curve

$$\widetilde{E}: \quad P_3(t)Y^2 = P_3(X).$$

On this curve, $(t, 1)$ is an ℓ-division point. The points (ξ, η) on $\widetilde{E}$ correspond to points $(\xi, \sqrt{P_3(t)}\eta)$ on E. Therefore the Frobenius automorphism $\phi : (x, y) \mapsto (x^q, y^q)$ translates to $(\xi, \eta) \mapsto (\xi^q, P_3(t)^{(q-1)/2}\eta^q)$ on $\widetilde{E}$.

There exist standard algorithms to determine an irreducible factor F of Ψ_ℓ; essentially one has to calculate the greatest common divisor of $T^{q^r} - T$ and $\Psi_\ell(T)$ for $r = 1, 2, \ldots$. However these algorithms are too expensive compared with all other operations, so it is better to leave Ψ_ℓ unfactorized and work over the ring $R := \mathbb{F}_q[T]/(\Psi_\ell(T))$, which amounts to working simultaneously over all fields $\mathbb{F}_q[T]/(F_j(T))$, where F_j are the irreducible factors of Ψ_ℓ. Working with the ring R instead of a field can cause only problems when inverses of elements $\xi \neq 0$ have to be calculated. The calculation of an inverse is done using the extended Euclidean algorithm. If the inverse does not exist, one detects automatically a factor G of Ψ_ℓ. Hence this does not hurt but is rather useful because we can pass to the smaller ring $R' = \mathbb{F}_q[T]/(G(T))$.

The algorithm of Schoof we sketched so far was the first algorithm of polynomial complexity for the point counting problem on elliptic curves. However it is still too slow for the curves used in cryptography. Atkin, Elkies and others have contributed improvements, which make the algorithm practical. In the next section we will describe one such improvement.

2.7 Elkies Primes. As before let E be an elliptic curve defined over a finite field $\mathbb{F}_q$ of characteristic $p > 3$, with trace $c = (q+1) - \#E(\mathbb{F}_q)$ and let $\ell < p$ be a an odd prime. Recall that the Frobenius automorphism restricted to the two dimensional $\mathbb{F}_\ell$-vectorspace $E[\ell]$ of ℓ-division points of E satisfies the quadratic equation

$$\phi^2 - c_\ell \phi + q_\ell = 0,$$

where $c_\ell = c \bmod \ell$ and $q_\ell = q \bmod \ell$. Therefore the eigenvalues of $\phi \,|\, E[\ell]$ are

$$\lambda_{1,2} = \tfrac{1}{2}(c_\ell \pm \sqrt{c_\ell^2 - 4q_\ell}).$$

If $c_\ell^2 - 4q_\ell$ is a square in $\mathbb{F}_\ell$, which will be the case for about half of the primes ℓ, these eigenvalues belong to the field $\mathbb{F}_\ell$. Primes with this property are called Elkies primes for the given elliptic curve. For such primes an eigenvector of ϕ spans a 1-dimensional subspace $C \subset E[\ell]$ invariant under the Frobenius automorphism. C is a cyclic subgroup of E of order ℓ defined over the ground field $\mathbb{F}_q$, hence the isogeny $E \to E/C$ is also defined over $\mathbb{F}_q$. Furthermore

$$G(T) := \prod_{P \in (C \smallsetminus 0)/\pm 1} (T - x(P)) \in \mathbb{F}_q[T]$$

is a factor of degree $(\ell - 1)/2$ of the division polynomial $\Psi_\ell(T)$. The important thing about Elkies primes is that they can be determined without having to work explicitly in $E[\ell]$. This is done using the modular polynomials $\Phi_\ell(x, y)$, cf. [5], Chap. 11.9. These are polynomials of degree $\ell + 1$ with integer coefficients, hence they can also be regarded as polynomials over $\mathbb{F}_q$. If $j(E)$ is the j-invariant of the elliptic curve E then the zeroes of $\Phi_\ell(j(E), y)$ are the j-invariants of curves E/C, where C runs through the cyclic subgroups of E of order ℓ. Therefore ℓ is an Elkies prime if and only if the polynomial $\Phi_\ell(j(E), y) \in \mathbb{F}_q[y]$ has a zero in $\mathbb{F}_q$; this can be checked by computing the greatest common divisor of this polynomial and $y^q - y$. When a solution $j' \in \mathbb{F}_q$ of $\Phi_\ell(j(E), j') = 0$ has been found, there is also a procedure to calculate directly the factor $G(T)$ of the division polynomial $\Psi_\ell(T)$. With this, a substantial gain in efficiency of Schoof's point counting algorithm is achieved, because for the elliptic curves used in cryptography primes ℓ up to 100 or higher are needed, so it makes a big difference whether one has to deal with polynomials of degree $(\ell - 1)/2$ or $(\ell^2 - 1)/2$. There exist still further improvements, for example replacing the modular polynomials Φ_ℓ, whose coefficients grow rapidly with ℓ, by simpler polynomials. We refer to Blake/Seroussi/Smart [2] and the references given there. We have restricted our attention here to elliptic curves over finite fields with large prime characteristic. For curves over fields of characteristic 2, other methods exist.

Bibliography

1. Atkin, A.O.L., Morain, F.: Elliptic curves and primality proving. *Math. Comp.* **61**, 29-67 (1993).

2. Blake, I., Seroussi, G., Smart, N.: *Elliptic Curves in Cryptography*. LMS Lecture Notes Series 265, Cambridge University Press, 1999.

3. Cohen, H.: *A Course in Computational Algebraic Number Theory*. Springer-Verlag, 1996.

4. Goldwasser, S., Kilian, J.: Almost all primes can be quickly certified. *18th STOC*, 316-329 (1986).

5. Husemöller, D.: Elliptic Curves. 2^{nd} edition, Springer-Verlag, to appear.

6. Lenstra, H.W.: Factoring integers with elliptic curves. *Ann. Math.* **126**, 649-673 (1987).

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

8. Schoof, R.: Counting points on elliptic curves over finite fields. *J. Théorie des Nombres de Bordeaux* **7**, 219-254 (1995).

9. Solovay, R., Strassen, V.: A fast Monte Carlo test for primality. *SIAM J. Comp.* **6**, 84-85 (1977). Erratum Vol. **7**, 118 (1978).

Otto Forster, Math. Institut der LMU München (Germany)
Email: `forster@mathematik.uni-muenchen.de`