Counting the points on an elliptic curve on a low-memory device

János A. Csirik

University of California, Berkeley

1. Goals, performance
2. Overview
3. Elliptic curve calculations
4. Modular curve calculations
5. Concluding remarks

1. Goals, performance

The main goal of this talk is *advertising*. (No surprising new results here.)

The aim of the algorithm is to produce a “random” elliptic curve E over $\mathbb{F}_q$, with a point P of order a large prime r.

We want to conserve memory for possible use in small widgets.

Choice of q: we take a 200-bit prime p. (Alternatives: 2^n, or p^n with a 32-bit prime p.)

Finding P is easy once we get some E with $\#E(\mathbb{F}_p) = xr$ with x small.

Performance: it takes about 160 kilobytes and 1 day. (test on P133, several curves at once)

2. Overview

Schoof–Elkies–Atkin algorithm

R. Schoof, N.D. Elkies, A.O.L. Atkin, F. Morain

Given E defined over $\mathbb{F}_p$,

$$\#E(\mathbb{F}_p) = p + 1 - t.$$

By Hasse's theorem, $t \leq 2\sqrt{p}$.

Determine $t \bmod \ell$ for a bunch of ℓ, then use the Chinese Remainder Theorem.

Try lots of ℓ. This method will work for about half of them. Trying 36 different primes for ℓ in the range 3..191, we are able to determine t for more than half of all E. Expected product of auxiliary primes ℓ that work $= 7.8 \times 10^{30}$, whereas $4\sqrt{p} < 6 \times 10^{30}$.

3. Elliptic curve calculations

Let ℓ be any prime other than p. Let

$$E \quad : \quad Y^2 = X^3 + a_4 X + a_6.$$

Let $E[\ell]$ denote the ℓ-torsion points of E. Then $E[\ell]$ is a two-dimensional vector space over $\mathbb{F}_\ell$.

The *Frobenius* map $F : (x, y) \mapsto (x^p, y^p)$ acts as a linear map on $E[\ell]$. The determinant of F is p, the trace is t. (We can consider t and p modulo ℓ here.) By the Cayley-Hamilton theorem,

$$F^2 - tF + p$$

acts as 0 on the vector space $E[\ell]$. Since t modulo ℓ is the only unknown thing here, we can try each of the ℓ possibilities to see which one is correct. This is Schoof's algorithm.

If the action of F has eigenvalues of $\mathbb{F}_\ell$, there will be two eigenspaces. It is sufficient to determine one eigenvalue e, since the other one is p/e and then $t = e + p/e$. (Elkies, 1/2.)

Assume we know one of the eigenspaces B. Then we try which of the possible e modulo ℓ would make

$$FQ = eQ$$

for each $Q \in B$.

Let $h(X)$ be the polynomial such that $(x, y) \in B$ iff $h(x) = 0$. The above amounts to finding e such that

$$\begin{aligned}(X^p, Y^p) \;&\equiv\; e(X, Y)\\ &\qquad (\mathrm{mod}\ h(X), Y^2 - X^3 - a_4X - a_6).\end{aligned}$$

Pick some small primitive root s modulo ℓ. We can successively calculate (X, Y), $s(X, Y)$, $s^2(X, Y)$, . . . , and compare them to (X^p, Y^p) until we find a match.

Storage: 10 polynomials modulo $h(X) = 850$ elements of $\mathbb{F}_p = 25$ kilobytes. Running the process for a given E for all good ℓ requires about 40 minutes.

4. Modular curve calculations

If E is an elliptic curve and B is a subgroup of order ℓ, we get an *isogeny* $E \to E/B$ of degree ℓ. Every isogeny arises in this way. ($\ell + 1$ of them)

The isogeny is defined over $\mathbb{F}_p$ iff B is an eigenspace for F iff $j(E/B) \in \mathbb{F}_p$. Therefore to carry out the elliptic curve calculation, we just need the polynomial $h(X)$ associated to the kernel of an isogeny $E \to E_1$ that is defined over $\mathbb{F}_p$.

The isogenies $E_0 \to E_1$ of degree ℓ correspond to points on the curve $Y_0(\ell)$. The isogenies defined over $\mathbb{F}_p$ correspond to points defined over $\mathbb{F}_p$. So we need to write down an equation for $Y_0(\ell)$ and find the $\ell+1$ points that correspond to isogenies originating from $E = E_0$.

The curve $Y_0(\ell)$ has various functions on it, like $j = J(E_0)$ and $\tilde{\jmath} = j(E_1)$. Use these and more complicated functions to read off the $\ell+1$ possible polynomials $h(X)$.

The curve $Y_0(\ell)$ is a certain quotient of the upper half plane, so functions on it can be written as q-expansions with $q = \exp(2\pi i\tau)$. For example,

$$
\begin{aligned}
j &= j(E_0) = q^{-1} + 744 + 196884q + \cdots, \\
\tilde{\jmath} &= j(E_1) = q^{-\ell} + 744 + 196884q^{\ell} + \cdots.
\end{aligned}
$$

We come up (using Atkin trick) with a certain function f on $Y_0(\ell)$ and use the q-expansions of f and j to write down a polynomial relation $\Psi_\ell(f, j) = 0$. We can choose f in such a way that also $\Psi_\ell(f, \tilde{\jmath}) = 0$.

Starting with $j \in \mathbb{F}_p$, we find all roots f in $\mathbb{F}_p$ of $\Psi_\ell(F, j) = 0$. For each of those, we find all roots $\tilde{\jmath}$ in $\mathbb{F}_p$ of $\Psi_\ell(f, \tilde{J}) = 0$. Once we have $j, f, \tilde{\jmath}$, we know enough of the isogeny to determine $h(X)$.

To determine Ψ_ℓ, we need to calculate up to 1400 terms in f. The total storage requirement of this step is about 160 kilobytes (5700 elements of $\mathbb{F}_p$). The calculation of all the Ψ_ℓ takes about 1 hour. (choice of the ℓ)

5. Concluding remarks

Given E and Ψ_ℓ we can calculate a specific array of 5 elements of $\mathbb{F}_p$ which can then be used to determine $h(X)$. Therefore, the total memory use is essentially the *maximum* of the memory uses of the elliptic curve and the modular curve parts.

Here is a list of possible improvements.

- Factoring $h(X)$
- Dewaghe's improvement
- Kangaroo traps when there are few possible t
- Using more memory when finding e
- Isogeny cycles by Couveignes–Dewaghe–Morain