18.783 Elliptic Curves
Lecture 8

Andrew Sutherland

February 23, 2022

Schoof's algorithm

In 1985 René Schoof introduced a polynomial-time algorithm for computing $\#E(\mathbb{F}_q)$. Schoof's strategy is to compute the trace of Frobenius modulo many small primes ℓ.

Algorithm

Given an elliptic curve E over a finite field $\mathbb{F}_q$ compute $\#E(\mathbb{F}_q)$ as follows:

1. Initialize $M \leftarrow 1$ and $t \leftarrow 0$.
2. While $M \le 4\sqrt{q}$, for increasing primes $\ell = 2, 3, 5, \ldots$ that do not divide q:
   - **2.1** Compute $t_\ell = \operatorname{tr}\pi \bmod \ell$.
   - **2.2** Set $t \leftarrow \left(M(M^{-1} \bmod \ell)t_\ell + \ell(\ell^{-1} \bmod M)t\right) \bmod \ell M$ and then $M \leftarrow \ell M$.
3. If $t > M/2$ then set $t \leftarrow t - M$.
4. Output $q + 1 - t$.

Step 2b uses an iterative CRT approach to ensure that $t \equiv \operatorname{tr}\pi_E \bmod M$ always holds. Hasse's theorem implies $t = \operatorname{tr}\pi_E$ after Step 3, so that $\#E(\mathbb{F}_q) = q + 1 - t$ in Step 4.

Preliminary complexity analysis

Let ℓ_{max} be the largest prime ℓ for which the algorithm computes t_ℓ.
The Prime Number Theorem (or even just Chebyshev's theorem) implies that

$$\sum_{\text{primes } \ell \leq x} \log \ell \sim x$$

as $x \to \infty$, and therefore

$$\ell_{max} \sim \log 4\sqrt{q} \sim \frac{1}{2}n = O(n),$$

where $n = \log q$, so we need $O(\frac{n}{\log n})$ primes ℓ.

The cost of Step 2.2 is bounded by $O(\mathsf{M}(n) \log n)$, thus if we can compute t_ℓ in Step 2.1 in time bounded by a polynomial in n and ℓ, we have a polynomial-time algorithm.

If $f(n)$ is the cost of Step 2.1 the total complexity is $O(n\mathsf{M}(n) + nf(n)/\log n)$.

Computing t_2

Assuming q is odd (which we do), $t = q + 1 - \#E(\mathbb{F}_q)$ is divisible by 2 if and only if $\#E(\mathbb{F}_q)$ is divisible by 2, equivalently, if and only if $E(\mathbb{F}_q)$ contains a point of order 2.

If E has Weierstrass equation $y^2 = f(x)$, then the points of order 2 in $E(\mathbb{F}_q)$ are precisely those of the form $(x_0, 0)$, where $x_0 \in \mathbb{F}_q$ is a root $f(x)$.

We can thus compute $t_2 := \operatorname{tr} \pi_E \bmod 2$ as

$$t_2 = \begin{cases} 0 & \text{if } \deg\Big(\gcd(f(x), x^q - x)\Big) > 0, \\ 1 & \text{otherwise.} \end{cases}$$

This is a deterministic computation (we need randomness to efficiently *find* the roots of $g(x)$, but we can efficiently *count* them deterministically). It takes $O(n\mathsf{M}(n))$ time.

The characteristic polynomial of the Frobenius endomorphism

The Frobenius endomorphism $\pi_E \in \mathrm{End}(E)$ satisfies its characteristic equation

$$\pi_E^2 - t\pi_E + q = 0,$$

with $t = \mathrm{tr}\,\pi$ and $q = \deg \pi$. Restricting to the ℓ-torsion subgroup $E[\ell]$ yields

$$\pi_\ell^2 - t_\ell \pi_\ell + q_\ell = 0, \tag{1}$$

which we view as an identity in $\mathrm{End}(E[\ell])$. Here $t_\ell \equiv t \bmod \ell$ and $q_\ell \equiv q \bmod \ell$ correspond to restrictions of the scalar multiplication endomorphisms $[t], [q] \in \mathrm{End}(E)$.

But we can also compute q_ℓ as

$$q_\ell = q_\ell \cdot [1]_\ell = [1]_\ell + \cdots + [1]_\ell$$

using double-and-add, provided that we know how to explicitly compute in $\mathrm{End}(E[\ell])$.

Computing the trace of Frobenius modulo ℓ

Our strategy to compute t_ℓ is simple: for $c = 0, 1, \ldots, \ell - 1$ compute

$$\pi_\ell^2 - c\pi_\ell + q_\ell$$

and check whether it is equal to 0 (as an element of $\operatorname{End}(E[\ell])$).

The following lemma shows that whenever this occurs we must have $c = t_\ell$.

Lemma

Let $E/\mathbb{F}_q$ be an elliptic curve with Frobenius endomorphism π, let ℓ be a prime not dividing q, and let $P \in E[\ell]$ be nonzero. Suppose that for some integer c the equation

$$\pi_\ell^2(P) - c\pi_\ell(P) + q_\ell(P) = 0$$

holds. Then $c \equiv t_\ell = \operatorname{tr}\pi \bmod \ell$.

Arithmetic in $\mathrm{End}(E[\ell])$ for odd primes ℓ

Let $h = \psi_\ell(x)$ be the ℓth division polynomial of $E\colon y^2 = f(x) = x^3 + Ax + B$, whose roots are the x-coordinates of the nonzero elements of $E[\ell]$. To represent elements of $\mathrm{End}(E[\ell])$ as rational maps, we work in the ring

$$\mathbb{F}_q\,[x,y]\,/(h(x), y^2 - f(x)).$$

We have

$$\begin{aligned}
\pi_\ell &= \Big(x^q \bmod h(x),\ y^q \bmod (h(x), y^2 - f(x))\Big)\\
&= \Big(x^q \bmod h(x),\ \Big(f(x)^{(q-1)/2} \bmod h(x)\Big)\,y\Big),\\
[1]_\ell &= \Big(x \bmod h(x),\ (1 \bmod h(x))\,y\Big)
\end{aligned}$$

We shall represent elements of $\mathrm{End}(E[\ell])$ in the form $(a(x), b(x)\,y)$, where $a, b \in \mathbb{F}_q[x]/(h(x))$ are uniquely represented as polynomials in $\mathbb{F}_q[x]$ reduced modulo h.

Multiplication in $\mathrm{End}(E[\ell])$

Given endomorphisms $\alpha_1, \alpha_2 \in \mathrm{End}(E[\ell])$ represented as

$$\alpha_1 = \big(a_1(x), b_1(x)y\big),$$
$$\alpha_2 = \big(a_2(x), b_2(x)y\big),$$

their product $\alpha_3 = \alpha_1\alpha_2$ in $\mathrm{End}(E[\ell])$ is the composition $\alpha_3 = \alpha_1 \circ \alpha_2$, which may we explicitly compute as

$$\begin{aligned}\alpha_3 &= \big(a_3(x), b_3(x)y\big)\\ &= \big(a_1(a_2(x)),\ b_1(a_2(x))b_2(x)\,y\big)\,,\end{aligned}$$

with $a_3(x)$ and $b_3(x)$ uniquely represented by their reductions modulo $h(x)$.

Addition in $\mathrm{End}(E)[\ell]$)

Given $\alpha_1 = (a_1(x), b_1(x)y)$, $\alpha_2 = (a_2(x), b_2(x)y)$, we want to compute $\alpha_3 = \alpha_1 + \alpha_2$.
For non-opposite affine points $(x_3, y_3) = (x_1, y_1) + (x_2, y_2)$ the group law on E tells us

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

$$m = \begin{cases} \frac{y_1 - y_2}{x_1 - x_2} & \text{if } x_1 \neq x_2, \\ \frac{3x_1^2 + A}{2y_1} & \text{if } x_1 = x_2. \end{cases}$$

Plugging in $x_1 = a_1(x),\ x_2 = a_2(x),\ y_1 = b_1(x)y,\ y_2 = b_2(x)y$, we obtain

$$m(x,y) = \begin{cases} \frac{b_1(x) - b_2(x)}{a_1(x) - a_2(x)} y = r(x)y & \text{if } x_1 \neq x_2, \\ \frac{3a_1(x)^2 + A}{2b_1(x)y} = \frac{3a_1(x)^2 + A}{2b_1(x)f(x)} y = r(x)y & \text{if } x_1 = x_2. \end{cases}$$

Now $m(x,y)^2 = (r(x)y)^2 = r(x)^2 f(x)$, so $\alpha_1 + \alpha_2 = \alpha_3 = (a_3(x),\ b_3(x)y)$ with

$$a_3 = r^2 f - a_1 - a_2, \qquad b_3 = r(a_1 - a_3) - b_1.$$

Dealing with zero divisors in $\mathbb{F}_q[x]/(h)$

If the denominator of $r = u/v$ is invertible in $\mathbb{F}_q[x]/(h(x))$ we can write $r = uv^{-1} \bmod h$ and put $\alpha_3 = (a_3(x), b_3(x)y)$ in our desired form, with $a_3, b_3 \in \mathbb{F}_q[x]/(h(x))$ uniquely represented as polynomials in $\mathbb{F}_q[x]$ reduced modulo h.

But this may not be possible! The ring $\mathbb{F}_q[x]/(h(x))$ is not necessarily a field.

At first glance this might appear to be a problem, but in fact it can only help us. If v is not invertible in $\mathbb{F}_q[x]/(h(x))$ then $\gcd(v, h)$ is a nontrivial factor of h (because we must have $\deg v < \deg h$).

Our strategy in this situation is to replace h by $g = \gcd(v, h)$ and compute t_ℓ by working in the smaller ring $\mathbb{F}_q[x]/(g(x))$. This will speed things up!

The lemma implies that we can restrict our attention to the action of π_ℓ on the subset of points $P \in E[\ell]$ whose x-coordinates are roots of $g(x)$.

Schoof's algorithm for computing the trace of Frobenius modulo ℓ

Algorithm

Given $E: y^2 = f(x)$ over $\mathbb{F}_q$ and an odd prime ℓ, compute t_ℓ as follows:

1. Compute the ℓth division polynomial $h = \psi_\ell \in \mathbb{F}_q[x]$ for E.
2. Compute $\pi_\ell = (x^q \bmod h,\ (f^{(q-1)/2} \bmod h)y)$ and $\pi_\ell^2 = \pi_\ell \circ \pi_\ell$.
3. Use scalar multiplication to compute $q_\ell = q_\ell[1]_\ell$, and then compute $\pi_\ell^2 + q_\ell$.
   (If a non-invertible denominator arises, update h and return to step 2).
4. Compute $0, \pi_l, 2\pi_l, 3\pi_l, \ldots, c\pi_\ell$, until $c\pi_l = \pi_l^2 + q_l$.
   (If a non-invertible denominator arises, update h and return to step 2).
5. Output $t_\ell = c$.

An implementation of this algorithm can be found in this Sage worksheet.

A few final remarks

- Factors of $h(x)$ necessarily arise when E admits a rational ℓ-isogeny. Elkies optimization of Schoof's algorithm exploits this fact, allowing us to work with polynomials of degree $(\ell-1)/2$ rather than $(\ell^2-1)/2$.
- Additional optimizations due to Atkin in the case where E does not admit a rational ℓ-isogeny lead the Schoof-Elkies-Atkin (SEA) algorithm.
- For cryptographic size primes the SEA algorithm a few seconds (or less). The current SEA record is a 16,000-bit prime, far beyond the cryptographic range.
- Even Schoof's original algorithm can handle cryptographic size primes, but this was not widely recognized in the 1980's.
- Schoof's algorithm can be used to deterministically compute square roots of a fixed integer modulo a prime. This application was the motivation for Schoof's original paper.