

# 18.783 Elliptic Curves

## Lecture 8

Andrew Sutherland

September 30, 2025

# 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  $\ell$ .

## 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  $M \leq 4\sqrt{q}$ , for increasing primes  $\ell = 2, 3, 5, \dots$  that do not divide  $q$ :
  - 2.1 Compute  $t_\ell = \text{tr } \pi \pmod{\ell}$ .
  - 2.2 Set  $t \leftarrow \left( M(M^{-1} \pmod{\ell})t_\ell + \ell(\ell^{-1} \pmod{M})t \right) \pmod{\ell M}$  and then  $M \leftarrow \ell M$ .
3. If  $t > M/2$  then set  $t \leftarrow t - M$ .
4. Output  $q + 1 - t$ .

Step 2.2 uses an iterative CRT approach to ensure that  $t \equiv \text{tr } \pi_E \pmod{M}$  always holds. Hasse's theorem implies  $t = \text{tr } \pi_E$  after Step 3, so that  $\#E(\mathbb{F}_q) = q + 1 - t$  in Step 4.

## Preliminary complexity analysis

Let  $\ell_{\max}$  be the largest prime  $\ell$  for which the algorithm computes  $t_\ell$ .

The Prime Number Theorem (or even just Chebyshev's theorem) implies that

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

as  $x \rightarrow \infty$ , and therefore

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

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

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

If  $f(n)$  is the cost of Step 2.1, the total complexity is  $O(nM(n) + nf(n)/\log n)$ .

## 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 of  $f(x)$ .

We can thus compute  $t_2 := \text{tr } \pi_E \pmod{2}$  as

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

This is a deterministic computation (we need randomness to efficiently *find* the roots of  $f$ , but we can efficiently *count* them deterministically). It takes  $O(nM(n))$  time.

# The characteristic polynomial of the Frobenius endomorphism

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

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

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

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

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

But we can also compute  $q_\ell$  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  $\text{End}(E[\ell])$ .

## Computing the trace of Frobenius modulo $\ell$

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

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

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

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  $\pi$ , let  $\ell$  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  $c \equiv t_\ell = \text{tr } \pi \pmod{\ell}$ .*

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

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

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

We have

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

We shall represent elements of  $\text{End}(E[\ell])$  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 $\text{End}(E[\ell])$

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

$$\alpha_1 = (a_1(x), b_1(x)y),$$

$$\alpha_2 = (a_2(x), b_2(x)y),$$

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

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

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

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

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

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

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

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

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

## 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  $t_\ell$  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  $\pi_\ell$  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 $\ell$

## Algorithm

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

1. Compute the  $\ell$ th 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  $0, \pi_\ell, 2\pi_\ell, 3\pi_\ell, \dots, c\pi_\ell$ , until  $c\pi_\ell = \pi_\ell^2 + q_\ell$ .  
(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  $\ell$ -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  $\ell$ -isogeny lead to the Schoof–Elkies–Atkin (SEA) algorithm.
- For cryptographic-size primes the SEA algorithm takes 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 1980s.
- 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](#).

MIT OpenCourseWare

<https://ocw.mit.edu>

18.783 / 18.7831 Elliptic Curves

Fall 2025

For information about citing these materials or our Terms of Use, visit: <https://ocw.mit.edu/terms>.