René Schoof's Algorithm for Computing $\#E(\mathbb{F}_p)$ for an elliptic curve $E : y^2 \equiv x^3 + A \cdot x + B \pmod{p}$

John McGee Radford University 10-Aug-2007

"A four-year-old child could understand that.
Run out and find me a four-year-old child, I can't make head or tail out of it."
- Groucho Marx (Duck Soup-1933)

René Schoof's 1985 paper entitled "Elliptic curves over finite fields and the computation of square roots mod $p$", details a polynomial time algorithm for determining $\#E(\mathbb{F}_p)$ [3]. The following steps outline Schoof's method.

Let $E$ be an elliptic curve over $\mathbb{F}_p$ given by

(1) $E : y^2 = x^3 + A \cdot x + B$, where $A, B \in \mathbb{F}_p$.

Hasse's Theorem tells us that the cardinality of the group of points is

(2) $\#E(\mathbb{F}_p) = p + 1 - t$, for some $t$ with $|t| \leq 2 \sqrt{q}$.

Let $\phi_p : E(\overline{\mathbb{F}}_p) \rightarrow E(\overline{\mathbb{F}}_p)$ such that $\phi_p((x, y)) = (x^p, y^p)$. Note that this is map of points with coordinates in the algebraic closure of $\mathbb{F}_p$. Then $\phi_p$ is an endomorphism called the Frobenius map. It has the following property, crucial to Schoof's algorithm [5]

(3) $\phi_p^2 - t \phi_p + p = 0 \ \forall \ P \in E(\overline{\mathbb{F}}_q)$

We can use (3) to compute $t \pmod{p_i}$ for a set of $L$ primes $l_1, l_2, \ldots, l_L$ such that

(4) $K = \prod_{i=1}^{L} l_i > 4 \sqrt{p}$

The Chinese Remainder Theorem is then applied to the resulting set of congruences to compute the unique

$t \pmod{K}$ such that $|t| \leq 2 \sqrt{q}$.

The order of the group is then given by $\#E(\mathbb{F}_q) = q + 1 - t$. Schoof showed that this algorithm will run time proportional to $\log^9 q$, based on analysis of the number of elementary operations required [1,4]. Details of the algorithm follow.
The division polynomials \( \psi_n \) of an elliptic curve \( E \) are elements of \( \mathbb{F}_p[x, y] \) with the property that \( \psi_n(x, y) = 0 \) if and only if \( (x, y) \in E[n] = \{ P \in E(\mathbb{F}_p) \mid n P = O \} \). These polynomials are defined recursively as follows [5]

\[
\begin{align*}
\psi_0 &= 0, \quad \psi_1 = 1, \quad \psi_2 = 2 y \\
\psi_3 &= 3 x^4 + 6 a x^2 + 12 b x - a^2 \\
\psi_4 &= 4 y(x^6 + 5 a x^4 + 20 b x^3 - 5 a^2 x^2 - 4 a b x - 8 b^2 - a^3) \\
\psi_{2n} &= \psi_n(\psi_{n+2} \psi_{n-1} - \psi_{n-2} \psi_{n+1}) \quad n \in \mathbb{Z}, \ n > 2 \\
\psi_{2n+1} &= \psi_{n+2} \psi_n^3 - \psi_{n+1}^2 \psi_n \quad n \in \mathbb{Z}, \ n > 1
\end{align*}
\]

The following polynomials, based on these division polynomials, are used in Schoof's algorithm. Note that during the execution of the algorithm all of the polynomial arithmetic takes place modulo \( \psi_1 \) for small primes \( l \). Note also that these polynomials turn out to be univariate in \( x \) only by computing modulo the relation \( y^2 = x^3 + a x + b \). The numbering refers to the equation numbers in [2]. The derivation of these polynomials [2] is based on the point multiplication formula (5).

\[
(5) \quad n P = \left( x - \frac{\psi_{n+1} \psi_{n-1}}{\psi_n^2}, \frac{\psi_{n+2} \psi_{n-1} - \psi_{n-2} \psi_{n+1}}{4 y \psi_n^3} \right)
\]

\[
\begin{align*}
\alpha &= \psi_{k+2} \psi_{k-1}^2 - \psi_{k-2} \psi_{k+1}^2 - 4 \psi_k^3 x^{\nu^2 + 1} \\
\beta &= 4 y \psi_k (\psi_k^2 (x - x^\nu) - \psi_{k-1} \psi_{k+1}) \\
p_{16}(x, y) &= (x^\nu - x) \psi_k^2 - \psi_{k-1} \psi_{k+1} \\
p_{17}(x, y) &= (x^\nu - x) \psi_w^2 - \psi_{w-1} \psi_{w+1} \\
p_{18}(x, y) &= 4 \psi_w^3 y^{\nu^2 + 1} - \psi_{w+2} \psi_{w-1}^2 - \psi_{w-2} \psi_{w+1}^2 \\
p_{19}(x, y) &= \psi_k^{\nu^2} (\beta^2 (\psi_{k-1} \psi_{k+1} - \psi_k^2 (x^{\nu^2} + x^\nu + x) + \alpha \psi_k^2)) + \psi_k^2 \beta^2 (\psi_{t-1} \psi_{t-1})^\nu \\
p_{19}(x, y) &= 4 y^\nu \psi_k^3 (\alpha \beta^2 (\psi_k^2 (2 x^{\nu^2} + x) - \psi_{k-1} \psi_{k+1}) - \psi_k^2 (x^3 + \beta^3 y^{\nu^2})) - \beta^3 \psi_k^2 (\psi_{t+2} \psi_{t-1}^2 - \psi_{t-2} \psi_{t+1}^2)^\nu
\end{align*}
\]

We can now gives the details of Schoof’s algorithm for \( E : y^2 = x^3 + a x + b \) over \( \mathbb{F}_p \) as follows.

1. If \( \text{gcd}(x^3 + a x + b, \ x^\nu - x) = 1 \) then \( t \equiv 0 \) (mod 2), else \( t \equiv 1 \) (mod 2)
2. Create a set of small primes $S = \{l_i\}$ such that $\prod_{i=1}^{L} l_i > 4\sqrt{p}$.

3. Compute the first $L + 2$ division polynomials $\psi_k$.

4. For each $l \in S$, compute $k \equiv p \pmod{l}$

5. If $\gcd(p_{16}, \psi_l) \neq 1$ then there exists $P \in E[l]$ such that $\phi_l^2 P = \pm k P$.

6. If $k$ is not a quadratic residue mod $l$, then $t \equiv 0 \pmod{l}$ else

7. Compute $w$ such that $w^2 \equiv k \pmod{l}$

8. If $\gcd(p_{17}, \psi_l) = 1$ then $t \equiv 0 \pmod{l}$, else

9. If $\gcd(p_{18}, \psi_l) \neq 1$ then $t \equiv 2w \pmod{l}$, else $t \equiv -2w \pmod{l}$.

10. else we are in case two

11. For each $\tau \leq (l + 1)/2$

12. If $\gcd(p_{19}, \psi_l) \neq 1$ then

13. $\phi_l^2 + k \equiv \pm \tau \pmod{l}$ for some point in $E[l]$ so we test

14. If $\gcd(p_{19}, \psi_l) \neq 1$ then $t \equiv \tau \pmod{l}$ else $t \equiv -\tau \pmod{l}$

15. Next $\tau$

16. Next $l$

17. At this point we have computed $t \pmod{l}$ for all $l_i \in S$,

18. so we can use the Chinese Remainder Theorem to compute

19. $T \equiv t \pmod{N}$ where $N = \prod_{i=1}^{L} l_i$.

20. If $T$ is within Hasse's bounds then $t = T$, else $t \equiv -T \pmod{N}$ and

21. $\# E(F_p) = p + 1 - t$.

This completes the description of Schoof's algorithm.

A version of this algorithm has been developed in *Mathematica* [2] and tested for elliptic curves over fields as large as $\mathbb{F}_p$ with $p \sim 10^{30}$. 
References


