Found Comput Math (2014) 14:285–297
DOI 10.1007/s10208-013-9181-9

FOUNDATIONS OF COMPUTATIONAL MATHEMATICS
The Journal of the Society for the Foundations of Computational Mathematics

On the Distribution of Atkin and Elkies Primes

Igor E. Shparlinski · Andrew V. Sutherland

Received: 18 February 2012 / Revised: 6 July 2013 / Accepted: 4 November 2013 /
Published online: 19 February 2014

Abstract Given an elliptic curve E over a finite field $\mathbb{F}_q$ of q elements, we say that an odd prime $\ell \nmid q$ is an Elkies prime for E if $t_E^2 - 4q$ is a square modulo ℓ, where $t_E = q+1-\#E(\mathbb{F}_q)$ and $\#E(\mathbb{F}_q)$ is the number of $\mathbb{F}_q$-rational points on E; otherwise, ℓ is called an Atkin prime. We show that there are asymptotically the same number of Atkin and Elkies primes $\ell < L$ on average over all curves E over $\mathbb{F}_q$, provided that $L \geq (\log q)^\varepsilon$ for any fixed $\varepsilon > 0$ and a sufficiently large q. We use this result to design and analyze a fast algorithm to generate random elliptic curves with $\#E(\mathbb{F}_p)$ prime, where p varies uniformly over primes in a given interval $[x, 2x]$.

Keywords Elkies prime · Elliptic curve · Character sum

Mathematical Subject Classification 11G07 · 11L40 · 11Y16

1 Introduction

Let $\mathbb{F}_q$ be a finite field of q elements. For an elliptic curve E over $\mathbb{F}_q$ we denote by $\#E(\mathbb{F}_q)$ the number of $\mathbb{F}_q$-rational points on E and define the *trace of Frobenius* $t_E = q+1-\#E(\mathbb{F}_q)$; see [2,27] for background on elliptic curves. We say that an odd prime $\ell \nmid q$ is an *Elkies prime* for E if $t_E^2 - 4q$ is a quadratic residue modulo ℓ;

Communicated by Andrew Odlyzko.

I. E. Shparlinski (✉)
Department of Pure Mathematics, University of New South Wales, Sydney, NSW 2052, Australia
e-mail: igor.shparlinski@unsw.edu.au

A. V. Sutherland
Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
e-mail: drew@math.mit.edu

otherwise, $\ell \nmid q$ is called an *Atkin prime*. For any elliptic curve over a finite field, one expects about the same number of Atkin and Elkies primes $\ell < L$ as $L \to \infty$.

These primes play a key role in the Schoof–Elkies–Atkin (SEA) algorithm (see [2, §17.2.2 and §17.2.5]), and their distribution affects the performance of this algorithm in a rather dramatic way. Thus we define $N_a(E; L)$ and $N_e(E; L)$ as the number of Atkin and Elkies primes ℓ in the dyadic interval $\ell \in [L, 2L]$ for an elliptic curve E over $\mathbb{F}_q$, respectively. We clearly have

$$N_a(E; L) + N_e(E; L) = \pi(2L) - \pi(L) + O(1),$$

where $\pi(L)$ denotes the number of primes $\ell < L$, and one expects that

$$N_a(E; L) \sim N_e(E; L) \sim \frac{1}{2}(\pi(2L) - \pi(L)) \tag{1}$$

as $L \to \infty$.

Under the generalized Riemann hypothesis (GRH), using the bound of quadratic characters over primes, it was noted by Galbraith and Satoh that (1) holds for $L \geq (\log q)^{2+\varepsilon}$ for any fixed $\varepsilon > 0$ and $q \to \infty$; see [23, Appendix A], and also [12, Proposition 5.25] or [21, Ex. 5.a in §13.1]. However, the unconditional results are much weaker and essentially rely on our knowledge of the distribution of primes in arithmetic progressions; see [12, §5.9] or [21, Chapter 4 and 11]. In the opposite direction, it is shown in [26] that one must take at least $L \geq c \log p \log\log\log p$ for infinitely many primes p, where $c > 0$ is an absolute constant.

Here, we study the values of $N_a(E; L)$ and $N_e(E; L)$ on average over all elliptic curves E over $\mathbb{F}_q$. Let $\mathcal{E}_q$ be any set of representative of all isomorphism classes of elliptic curves over $\mathbb{F}_q$.

Theorem 1 *For any integer $\nu \geq 1$, we have*

$$\frac{1}{\#\mathcal{E}_q} \sum_{E \in \mathcal{E}_q} \left| N_*(E; L) - \frac{1}{2}(\pi(2L) - \pi(L)) \right|^{2\nu}$$
$$= O\left(\pi(2L)^{\nu} \log q \log\log q + \pi(2L)^{2\nu} q^{-1/2} L^{\nu} \log L\right),$$

where $N_*(E; L)$ is either $N_a(E; L)$ or $N_e(E; L)$.

For an appropriate choice of ν we obtain from Theorem 1 a nontrivial result in the range

$$(\log q)^{\varepsilon} \leq L \leq q^{1/2} (\log q)^{-1/2-\varepsilon}$$

for any fixed $\varepsilon > 0$ and all sufficiently large q. This range includes values of L that are much smaller than those addressed by the result of Galbraith and Satoh for any particular elliptic curve, even under the GRH.

In many applications it is more convenient to consider curves given by the family of short Weierstraß equations

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

where a and b run through $\mathbb{F}_q$, with $\gcd(q, 6) = 1$, and satisfy $4a^3 + 27b^2 \neq 0$. Since there are $O(q)$ pairs $(a, b) \in \mathbb{F}_q^2$ for which $E_{a,b}$ lies in a given isomorphism class, we easily derive from Theorem 1 the following result.

Corollary 2 *For any real $\varepsilon > 0$ and integer $C \geq 1$, for a sufficiently large prime p and $p^{1/2}(\log p)^{-1/2-\varepsilon} \geq L \geq (\log p)^{\varepsilon}$ there are at most $p^2(\log p)^{-C}$ pairs $(a, b) \in \mathbb{F}_p^2$ for which $4a^3 + 27b^2 \neq 0$ and*

$$N_*(E; L) < \frac{1}{3}(\pi(2L) - \pi(L)),$$

where $N_*(E; L)$ is either $N_a(E; L)$ or $N_e(E; L)$.

As an application of Corollary 2, in Sect. 5 we present Algorithm 2, which efficiently generates a random elliptic curve of prime order. Given an integer $x > 3$, we seek a random and sufficiently uniformly distributed element of the set $T(x)$ of all triples (p, a, b), where p is a prime in the interval $[x, 2x]$, while a and b are elements of $\mathbb{F}_p$ for which the elliptic curve $E_{a,b}$ in (2) has a prime number of $\mathbb{F}_p$-rational points. This problem arises in cryptographic applications of elliptic curves, where one typically requires a curve with prime (or near prime) order but wishes to choose a curve that is otherwise as generic as possible.[1]

We show that the output and complexity of Algorithm 2 (Sect. 5) satisfy the following theorem.

Theorem 3 *Given a real number $x > 3$, Algorithm 2 outputs a prime $p \in [x, 2x]$, two elements $a, b \in \mathbb{F}_p$, and $N = \#E_{a,b}(\mathbb{F}_p)$, where p is uniformly distributed over primes in $[x, 2x]$ and the pair (a, b) is then uniformly distributed over pairs in $\mathbb{F}_p \times \mathbb{F}_p$ for which $E_{a,b}(\mathbb{F}_p)$ is prime. Assuming the GRH, the expected running time of Algorithm 2 is $O((\log x)^5(\log\log x)^3 \log\log\log x)$.*

2 Preparations

We recall the notations $U = O(V)$, $V = \Omega(U)$, $U \ll V$, and $V \gg U$, all of which are equivalent to the statement that the inequality $|U| \leq cV$ holds asymptotically, with some constant $c > 0$. We also write $U = \widetilde{O}(V)$ to indicate that $|U| \leq V(\log V)^{O(1)}$. Throughout the paper, any implied constants in these symbols may occasionally depend, where obvious, on the integer parameter $\nu \geq 1$ and the real parameter $\varepsilon > 0$, and are absolute otherwise. We always assume that ℓ runs through the prime integers. We also assume that for a prime p the field $\mathbb{F}_p$ is represented by the integers in the interval $[0, p - 1]$.

[1] Some cryptographic security standards specifically preclude the use of alternative approaches such as the CM method for this very reason [18].

Let us first recall some known facts about elliptic curves, which are conveniently summarized by Lenstra [15]. In particular, we need the following well-known asymptotic estimate on the cardinality of $\#\mathcal{E}_q$; see [15, §1.4] for $\gcd(q,6)=1$, [11, Theorem 3.18] for $2 \mid q$, and [13] for $3 \mid q$.

Lemma 4 *We have*

$$\#\mathcal{E}_q = 2q + O(1).$$

Furthermore, let $f_q(t)$ be the number of isomorphism classes of curves E over $\mathbb{F}_q$, with $t_E = t$. McKee [20, Theorem 2] gives a more precise form of the upper bounds of Lenstra [15, Proposition 1.9] on $f_q(t)$, which we formulate together with the Hasse estimate on possible values of t; see [15, Proposition 1.5] or [2,27].

Lemma 5 *We have*

$$f_q(t) \ll \begin{cases} 0, & \text{if } |t| > 2q^{1/2}, \\ q^{1/2}\log q \log\log q, & \text{if } |t| \le 2q^{1/2}. \end{cases}$$

We also need some results on multiplicative character sums. More precisely, we concentrate on the sums of Jacobi symbols (a/b); see [12, § 3.5]. Let us first consider complete sums.

Lemma 6 *For any integer a and a product $m = \ell_1 \dots \ell_s$ of s distinct odd primes $\ell_1, \dots, \ell_s$ with* $\gcd(a,m)=1$ *we have*

$$\left|\sum_{t=0}^{m-1} \left(\frac{t^2-a}{m}\right)\right| = 1.$$

Proof We use the following special case of the well-known identity for sums of Legendre symbols with quadratic polynomials ([17, Theorem 5.48]):

$$\sum_{t=0}^{\ell-1} \left(\frac{t^2-a}{\ell}\right) = -1$$

for any prime $\ell \nmid a$. Applying the multiplicativity of complete character sums, see [12, Eq. 12.21], completes the proof. □

The following estimate is a slight generalization of [19, Lemma 2.2].

Lemma 7 *For any integers a and $T \ge 1$ and a product $m = \ell_1 \dots \ell_s$ of $s \ge 0$ distinct odd primes $\ell_1, \dots, \ell_s$ with* $\gcd(a,m)=1$ *we have*

$$\sum_{|t| \le T} \left(\frac{t^2-a}{m}\right) \ll T/m + C^s m^{1/2} \log m$$

for some absolute constant $C \ge 1$.

Proof The result is trivial when $s = 0$, that is, when $m = 1$.

For $s \geq 1$, as in [19], we note that the Weil bound applied to the mixed sums of additive and multiplicative characters with polynomials, of the type given in [12, Eq. 11.43], and the multiplicativity of complete character sums (see [12, Eq. 12.21]) imply that

$$\sum_{t=1}^{m} \left(\frac{t^2 - a}{m}\right) \exp\left(2\pi i \frac{\lambda t}{m}\right) \ll C^s m^{1/2}$$

holds for any integer λ and some absolute constant $C \geq 1$. Using the standard reduction between complete and incomplete sums (see [12, § 12.2]), we derive that for any integer K and any positive integer $L \leq m$ we have

$$\sum_{t=K+1}^{K+L} \left(\frac{t^2 - a}{m}\right) \exp\left(2\pi i \frac{\lambda t}{m}\right) \ll C^s m^{1/2} \log m. \tag{3}$$

Separating the summation range over t into $O(T/m)$ intervals of length m (and using Lemma 6 for the sums over these intervals) and at most one interval of length m (and using (3) for the sums over these intervals), we obtain the desired result. □

Finally, for any integer n we denote by $\omega_L(n)$ the number of primes in the interval $[L, 2L]$ that divide n.

Lemma 8 *For $L \geq 3$ and any integer $\nu \geq 1$ we have*

$$\sum_{|t|<T} \omega_L^{\nu}(t^2 - a) \ll \frac{T}{\log L} + \frac{L^{\nu}}{(\log L)^{\nu}}.$$

Proof We have

$$\sum_{|t|<T} \omega_L^{\nu}(t^2 - a) = \sum_{|t|<T} \left(\sum_{\substack{L \leq \ell \leq 2L \\ \ell \mid t^2 - a}} 1 \right)^{\nu} = \sum_{L \leq \ell_1, \dots \ell_\nu \leq 2L} \sum_{\substack{|t|<T \\ \mathrm{lcm}\,[\ell_1, \dots, \ell_\nu] \mid t^2 - a}} 1.$$

By the Chinese remainder theorem, for any squarefree $m \geq 1$ we have

$$\sum_{\substack{|t|<T \\ m \mid t^2 - a}} 1 \ll (T/m + 1) \sum_{\substack{t=1 \\ m \mid t^2 - a}}^{m} 1 \ll 2^{\omega(m)}(T/m + 1),$$

where $\omega(m)$ is the number of distinct prime divisors of m. Now, for each $j = 1, \dots, \nu$ we collect together the terms such that among $\ell_1, \dots \ell_\nu < L$, only j are distinct. We then obtain

$$\sum_{|t|<T} \omega_L^\nu(t^2-a) \ll \sum_{j=1}^{\nu} \sum_{L\le \ell_1,\dots,\ell_j\le 2L} \left(\frac{T}{\ell_1\dots\ell_j}+1\right)$$
$$\le \sum_{j=1}^{\nu}\left(T\left(\sum_{L\le\ell\le 2L}\frac{1}{L}\right)^j + \pi(2L)^j\right).$$

Applying the prime number theorem completes the proof. □

3 Proof of Theorem 1

Clearly, we have

$$N_a(E;L)-N_e(E;L)=\sum_{L\le\ell\le 2L}\left(\frac{t_E^2-4q}{\ell}\right)+O\left(\omega_L(t_E^2-4q)+1\right),$$

where, as before, $\omega_L(n)$ denotes the number of primes $\ell\in[L,2L]$ with $\ell \mid n$.

Therefore,

$$\frac{1}{\#\mathcal{E}_q}\sum_{E\in\mathcal{E}_q}\left|N_*(E;L)-\frac{1}{2}\left(\pi(2L)-\pi(L)\right)\right|^{2\nu} \ll \frac{1}{\#\mathcal{E}_q}U_\nu+\frac{1}{\#\mathcal{E}_q}V_\nu+1, \qquad (4)$$

where, as before, $N_*(E;L)$ is either $N_a(E;L)$ or $N_e(E;L)$ and

$$U_\nu=\sum_{E\in\mathcal{E}_q}\left|\sum_{L\le\ell\le 2L}\left(\frac{t_E^2-4q}{\ell}\right)\right|^{2\nu} \quad\text{and}\quad V_\nu=\sum_{E\in\mathcal{E}_q}\omega_L(4q-t_E^2)^{2\nu}.$$

By Lemma 5,

$$U_\nu=\sum_{|t|<2q^{1/2}} f_q(t)\left|\sum_{L\le\ell\le 2L}\left(\frac{t^2-4q}{\ell}\right)\right|^{2\nu}$$
$$\ll q^{1/2}\log q\log\log q\sum_{|t|<2q^{1/2}}\left|\sum_{L\le\ell\le 2L}\left(\frac{t^2-4q}{\ell}\right)\right|^{2\nu}, \qquad (5)$$

where $f_q(t)$ is defined as in Sect. 2. Furthermore,

$$\sum_{|t|<2q^{1/2}}\left|\sum_{L\le\ell\le 2L}\left(\frac{t^2-4q}{\ell}\right)\right|^{2\nu} = \sum_{3\le\ell_1,\dots,\ell_{2\nu}\le L}\ \sum_{|t|<2q^{1/2}}\left(\frac{t^2-4q}{\ell_1\dots\ell_{2\nu}}\right).$$

For every $j = 0, \dots \nu$ let $\mathcal{Q}_j$ be the set of 2ν tuples $(\ell_1, \dots, \ell_{2\nu})$ of primes with $L \le \ell_1, \dots, \ell_{2\nu} \le 2L$ such that the product $r = \ell_1 \dots \ell_{2\nu}$ is of the form $r = k^2 m$, with m squarefree and k the product of j primes.

For the cardinalities of these sets we clearly have

$$\#\mathcal{Q}_j \ll (\pi(2L) - \pi(L))^{2\nu - j} \ll \frac{L^{2\nu-j}}{(\log L)^{2\nu-j}}.$$

Using Lemma 7 for $(\ell_1, \dots, \ell_{2\nu}) \in \mathcal{Q}_j$, $j = 0, \dots, \nu$, we obtain

$$\begin{aligned}\sum_{|t|<2q^{1/2}} \left| \sum_{L \le \ell \le 2L} \left(\frac{t^2 - 4q}{\ell} \right) \right|^{2\nu} &\ll \sum_{j=0}^{\nu} \#\mathcal{Q}_j (q^{1/2}/L^{2\nu-2j} + L^{\nu-j} \log L) \\ &\ll \sum_{j=0}^{\nu} \left(q^{1/2} \frac{L^j}{(\log L)^{2\nu-j}} + \frac{L^{3\nu-2j}}{(\log L)^{2\nu-j-1}} \right) \\ &\ll q^{1/2} \frac{L^\nu}{(\log L)^\nu} + \frac{L^{3\nu}}{(\log L)^{2\nu-1}}.\end{aligned}$$

Inserting this bound into (5) we obtain

$$U_\nu \ll \left(q^{1/2} \frac{L^\nu}{(\log L)^\nu} + \frac{L^{3\nu}}{(\log L)^{2\nu-1}} \right) q^{1/2} \log q \log\log q. \tag{6}$$

Finally, by Lemmas 5 and 8, we have

$$V_\nu \ll \left(\frac{q^{1/2}}{\log L} + \frac{L^{2\nu}}{(\log L)^{2\nu}} \right) q^{1/2} \log q \log\log q. \tag{7}$$

Substituting (6) and (7) into (4) and recalling Lemma 4, we conclude the proof.

4 Point Counting on Random Elliptic Curves

We now consider the problem of generating a random elliptic curve whose group of $\mathbb{F}_p$-rational points has prime order. One approach is to fix the prime p and then count points on randomly generated elliptic curves over $\mathbb{F}_p$ until a curve with prime order is found. Using the SEA point-counting algorithm, this procedure heuristically has an expected running time of $\widetilde{O}(n^5)$, where $n = \log p$. However, for a fixed prime p we cannot hope to prove even a polynomial time bound because even under the GRH the Hasse interval $[p - 2\sqrt{p}, p + 2\sqrt{p}]$ is too narrow to permit a useful lower bound on the number of primes it contains. Thus we let p vary over an interval $[x, 2x]$, which at least makes a polynomial-time bound feasible; see [14].

A second obstacle to obtaining an $\widetilde{O}(n^5)$ expected time bound is that the expected running time of the SEA algorithm is not known to be polynomial in n, unless we assume the GRH. Even with the GRH, the expected running time of the SEA algorithm on any particular curve is only bounded by $\widetilde{O}(n^5)$, yielding an $\widetilde{O}(n^6)$ bound overall. However, for randomly generated curves, Theorem 1 yields a tighter bound, on average, allowing us to prove an $\widetilde{O}(n^5)$ bound on the expected time to find a curve of prime order, under the GRH.

We first present an algorithm that attempts to count the points on the elliptic curve $E_{a,b}$ modulo p using a simplified version of the SEA algorithm that relies only on Elkies primes. In the course of doing so, the algorithm may discover that p is composite (using the Miller–Rabin algorithm [22]) or that the curve $E_{a,b}$ is singular modulo p; in either case, it outputs 0; otherwise, it returns a positive integer N in the Hasse interval $[p-2\sqrt{p}, p+2\sqrt{p}]$. If p is in fact prime (and $E_{a,b}$ is not singular), then N is equal to $\#E_{a,b}(\mathbb{F}_p)$.

Algorithm 1 below specifies a "background task" initiated in Step 1 that is meant to execute in parallel with the main task of the algorithm; this parallelism can easily be simulated by a traditional sequential algorithm on a single processor. The completion of either task terminates the algorithm.

Algorithm 1 *Point counting modulo p using Elkies primes.*

Input: *An integer $p > 3$ and integers $a, b \in [0, p-1]$.*
Output: *A positive integer $N \in [p-2\sqrt{p}, p+2\sqrt{p}]$ with $\#E_{a,b}(\mathbb{F}_p) = N$ if p is prime and $4a^3 + 27b^2 \not\equiv 0 \bmod p$, and 0 otherwise.*

1. In parallel with the steps below, repeatedly test p for compositeness using the Miller–Rabin algorithm [22]. If at any point p is found to be composite, then output 0 and terminate.
2. If $\gcd(4a^3 + 27b^2, p) \neq 1$, then output 0 and terminate. Otherwise, set $j \leftarrow 1728\frac{4a^3}{4a^3+27b^2} \bmod p$.
3. Test whether $E = E_{a,b} \bmod p$ is supersingular using Algorithm 2. If so, then output $p+1$ and terminate.
4. Set $i \leftarrow 0$, $M \leftarrow 1$, and for primes $\ell = 2, 3, 5, \ldots$, do the following:
   (a) Compute the modular polynomial $\Phi_\ell(X, Y)$ using Algorithm 1.
   (b) Compute $\phi(X) = \Phi_\ell(j, X)$ and $f(X) = \gcd(X^p - X, \phi(x))$ in the ring $(\mathbb{Z}/p\mathbb{Z})[X]$. If $\deg f = 0$, then proceed to the next prime ℓ.
   (c) Find a root $\tilde{j}$ of $f(X)$ modulo p.
   (d) Compute the Elkies polynomial $h(X)$ whose roots are the abscissae of the points in the kernel of the ℓ-isogeny ϕ from E to a curve $\widetilde{E}$ with j-invariant $\tilde{j}$.[2]
   (e) Using h, determine the integer $\lambda \in [1, \ell-1]$ for which the p-power Frobenius action on $\ker\phi$ is equivalent to multiplication by λ. If no such λ exists, then output 0 and terminate.
   (f) Set $i \leftarrow i+1$, $\ell_i \leftarrow \ell$, $M \leftarrow M\ell$, and $t_i \leftarrow \lambda + p/\lambda \bmod \ell$.
   (g) If $M > 4\sqrt{p}$, then proceed to Step 5. Otherwise, continue Step 4.
5. Compute the unique integer $t \in [-M, M]$ for which $t \equiv t_i \bmod \ell_i$ for each Elkies prime ℓ_i. If $|t| > 2\sqrt{p}$, then output 0 and terminate; otherwise, output $N = p+1 -t$ and terminate.

[2] The special case $(\partial\Phi_\ell/\partial X)(j, \tilde{j}) = (\partial\Phi_\ell/\partial Y)(j, \tilde{j}) = 0$ must be handled separately; see the proof of Lemma 9 for details.

We note that the algorithm is not in any sense required to be "correct" when p is composite; it may output either 0 or any integer N in the Hasse interval in this case; however, it is required to terminate with probability 1, and we want to tightly bound its expected running time in Lemma 9. This is the purpose of the Miller–Rabin tests begun in Step 1 of Algorithm 1, which are repeated continuously, in parallel with the remaining steps of the algorithm (as noted earlier, this "parallelism" may be simulated). The algorithm terminates whenever it either proves that p is composite or completes Step 5, whichever happens first. The reason for doing this is that if p is composite, we cannot necessarily prove polynomial time bounds on certain steps of the algorithm (in particular, the root-finding operation in Step 4c).

Assuming that p is prime, the value j computed in Step 2 is the j-invariant of the elliptic curve $E = E_{a,b}$ over $\mathbb{F}_p$. The classical modular polynomial Φ_ℓ parametrizes pairs of ℓ-isogenous elliptic curves; the roots of $\Phi_\ell(j(E), X)$ are the j-invariants of the curves $\widetilde{E}$ that are related to E by a cyclic isogeny of degree ℓ. There exists such an elliptic curve $\widetilde{E}$ defined over $\mathbb{F}_p$ precisely when ℓ is an Elkies prime for E; thus, Elkies and Atkin primes are distinguished in Steps 4b and c, which attempt to find a root of $\Phi_\ell(j(E), X)$ in $\mathbb{F}_p$. Steps 4c–f then apply the standard SEA procedure for computing the trace of Frobenius modulo an Elkies prime ℓ, as described by Schoof in [25].

We now consider the complexity of Algorithm 1. We use the asymptotic bound $O(n \log n \log\log n)$ of Schönhage and Strassen [24] to bound the time $\mathsf{M}(n)$ to multiply two n-bit integers (see also [9]) and note that all of our complexity estimates count bit operations.

Lemma 9 *Let $n = \lceil \log p \rceil$, and assume the GRH. For composite p, the expected running time of Algorithm 1 is $O(n^2 \log n \log\log n)$. For prime p, the average expected running time of Algorithm 1 over integers $a, b \in [0, p-1]$ is $O(n^4 (\log n)^2 \log\log n)$.*

Proof We expect to detect a composite p using $O(1)$ Miller–Rabin tests, each of which has complexity $O(n\mathsf{M}(n)) = O(n^2 \log n \log\log n)$, the time to perform an exponentiation modulo p. This proves the first claim.

We now assume p is prime. The complexity of Step 2 is $O(\mathsf{M}(n) \log n)$, and Step 3 runs in $O(n^3 \log n \log\log n)$ expected time; see [28, Proposition 4].

Let m be the largest prime ℓ used in Step 4. We have $\log M \geq n/2$; thus, by the prime number theorem, $m \gg n$. Ignoring constant factors, we may use m as an upper bound on both ℓ and n. Table 1 estimates the costs of Steps 4a–f in terms of ℓ and n and also gives bounds in terms of m. We use standard asymptotic bounds on the complexity of (fast) arithmetic operations in $\mathbb{Z}/p\mathbb{Z}$ and $\mathbb{Z}/p\mathbb{Z}[X]$, all of which can be found in [9].[3]

In Step 4a we use the isogeny volcano algorithm of [5] to compute the modular polynomial Φ_ℓ, and it is here that we need to assume the GRH. In the complexity bound for Step 4d we include the cost of computing and evaluating various partial derivatives of Φ_ℓ modulo p and use Elkies' algorithm to compute the kernel polynomial $h(X)$; see [7] and [8, Chapter 25] for details and [4] for further optimizations. In the

[3] Some of these bounds can be improved using Kronecker substitution to multiply polynomials in $\mathbb{Z}/p\mathbb{Z}[X]$, but this does not change the overall complexity.

Table 1 Complexity bounds for Step 4 of Algorithm 1

Step	Result	Expected time $O(\cdots)$	In terms of m
a	$\Phi_\ell(X, Y)$	$\ell^3(\log\ell)^3\log\log\ell$	$m^3(\log m)^3\log\log m$
b	$\phi(X)$	$\ell^2\mathsf{M}(\ell\log\ell+n)$	$m^3(\log m)^2\log\log m$
	$X^p \bmod \phi$	$n\mathsf{M}(\ell)\mathsf{M}(n)$	$m^3(\log m)^2(\log\log m)^2$
	$f(X)$	$\mathsf{M}(\ell)\mathsf{M}(n)\log\ell+\ell\mathsf{M}(n)\log n$	$m^2(\log m)^3(\log\log m)^2$
c	$\tilde{j}$	$\mathsf{M}(\ell)\mathsf{M}(n)n$	$m^3(\log m)^2(\log\log m)^2$
d	$h(X)$	$\ell^2\mathsf{M}(n)+\mathsf{M}(n)\ell\log n$	$m^3\log m\log\log m$
e	λ	$\mathsf{M}(\ell)\mathsf{M}(n)n+\mathsf{M}(\ell)\mathsf{M}(n)\ell$	$m^3(\log m)^2(\log\log m)^2$
f	t_i	$\ell\mathsf{M}(\log\ell)\log\log\ell$	$m\log m(\log\log m)^2$
	M	$\mathsf{M}(n+\log\ell)$	$m\log m\log\log m$

complexity bound for Step 4e, the first term bounds the time to compute the action of Frobenius on $\ker\phi$ (this involves computing X^p and Y^p modulo h and $E_{a,b}$), while the second term bounds the time to compute the action of multiplication by λ on $\ker\phi$ for every integer λ in $[1, \ell-1]$; see [10] for details and optimizations.

The cost of Steps 4a–f is dominated by the $O(m^3(\log m)^3\log\log m)$ cost of Step 4a, which also dominates the cost of Steps 2, 3, and 5, the last of which has complexity $O(\mathsf{M}(m)\log m)$. The number of iterations in Step 4 is at most $\pi(m) = O(m/\log m)$; thus, when p is prime, the total expected running time of Algorithm 1 is $O(m^4(\log m)^2\log\log m)$.

To address the special case $(\partial\Phi_\ell/\partial X)(j,\tilde{j}) = (\partial\Phi_\ell/\partial Y)(j,\tilde{j}) = 0$, we note that, as explained by Schoof in [25, pp. 248–249], there are then only $O(\ell^2)$ possible values for N. For $p > 229$ only one of these candidates satisfies Mestre's theorem [25, Theorem 3.2]. By multiplying random points on $E_{a,b}(\mathbb{F}_p)$ and its quadratic twist by each of the candidate values for N, we can uniquely determine N in $O(\ell^2 n\mathsf{M}(n)) = O(m^4\log m\log\log m)$ expected time, which is dominated by the bound we derived earlier [and for $p \le 229$ we can simply enumerate the elements of $E_{a,b}(\mathbb{F}_p)$ by brute force].

We now notice that by Corollary 2 (taken with $C = 3$) and the prime number theorem, we have $m \ll n$ for all but $O(p^2 n^{-3})$ pairs $(a, b) \in \mathbb{F}_p^2$ for which, by the result of Galbraith and Satoh [23, Appendix A], we have $m \ll n^3$.

Thus, if we average over all integers of a, b in $[0, p-1]$ for a fixed prime p, then the expected value of m is $O(n)$, which completes the proof. □

5 Proof of Theorem 3

The proof is based on the analysis of the following procedure.

Algorithm 2 *Generation of a random elliptic curve with a prime number of rational points over a finite field.*

Input: *A real* $x > 3$.
Output: *A prime* $p \in [x, 2x]$, $a, b \in \mathbb{F}_p$, *and* $N = \#E_{a,b}(\mathbb{F}_p)$ *prime.*

1. Pick a uniformly random integer p in the interval $[x, 2x]$.
2. Pick uniformly random integers $a, b \in [0, p-1]$ and apply Algorithm 1 to $E_{a,b} \bmod p$, obtaining N. If Algorithm 1 finds that p is composite or that $p|(4a^3 + 27b^2)$, then return to Step 1.
3. Apply $\lceil \log x \rceil$ Miller–Rabin tests to both p and N. If either p or N is found to be composite, then return to Step 1.
4. Determine the primality of p and N using a randomized Agrawal–Kayal–Saxena (AKS) algorithm [3]. If N and p are both prime, then output p, a, b, and N, and terminate. Otherwise, return to Step 1.

The Miller–Rabin algorithm [22] attempts to prove that a given integer p is not prime (that is, composite) via a sequence of independent random tests, each of which detects a composite p with a probability of at least 3/4. Thus, the probability that the algorithm reaches Step 4 when N is composite is less than $1/\log x$. The primality testing algorithm used in Step 4 is a randomized version of the Agrawal–Kayal–Saxena algorithm [1] due to Bernstein [3] and determines whether N is prime or composite in $O(n^{4+\varepsilon})$ expected time for any $\varepsilon > 0$.

We now set $n = \lceil \log x \rceil$ and show that the expected running time of Algorithm 2 is $O(n^5 (\log n)^3 \log\log n)$. Step 2 of Algorithm 2 calls Algorithm 1 with parameters (p, a, b), where p is an integer chosen uniformly at random from the interval $[x, 2x]$ and (a, b) are chosen uniformly at random from $\mathbb{F}_p \times \mathbb{F}_p$. As before, let $T(x)$ denote the set of triples (p, a, b) for which both p and $N = \#E_{a,b}(\mathbb{F}_p)$ are prime [and $p \nmid (4a^3 + 27b^2)$, which we assume throughout].

We first show that the cardinality of $T(x)$ satisfies

$$\#T(x) \gg \frac{x^3}{(\log x)^2 \log\log x}. \tag{8}$$

By [14, Lemma 1], the number of pairs of primes (p, N) with $x \le p \le 2x$ and $p - \sqrt{p} \le N \le p + \sqrt{p}$ is $\Omega(x^{3/2}/(\log x)^2)$. For each such pair (p, N), the number of pairs (a, b) with $0 \le a, b < p$ for which $\#E_{a,b}(\mathbb{F}_p) = N$ is $\frac{1}{2}(p-1)H(D)$, where $D = (p + 1 - N)^2 - 4p$, and $H(D)$ denotes the Hurwitz class number; see [6, Theorem 14.18]. Let $D = v^2 D_0$, where D_0 is a fundamental discriminant. By [29, Lemma 9], we have

$$H(D) \ge vH(D_0) \ge \frac{1}{3} v h(D_0),$$

and the GRH implies

$$h(D_0) \gg \sqrt{|D_0|}/\log\log|D_0|,$$

where $h(D_0)$ is the usual class number, by a theorem of Littlewood [16]. It follows that

$$H(D) \gg \sqrt{|D|}/\log\log|D| \gg \sqrt{x}/\log\log x$$

(see also comments in [15, §1.6]). Therefore, there are $\Omega(x^{3/2}/\log\log x)$ pairs (a, b) with $\#E_{a,b}(\mathbb{F}_p) = N$ and $\Omega(x^{3/2}/(\log x)^2)$ pairs of primes (p, N), which implies (8).

Thus, we expect to generate $O((\log x)^2 \log\log x) = O(n^2 \log n)$ random triples (p, a, b) in order to obtain a triple for which p and $N = \#E_{a,b}(\mathbb{F}_p)$ are both prime. Once this occurs, the algorithm successfully completes Steps 2–5 and terminates. We now consider the cost of processing each random triple, which we divide into three cases.

1. If p is composite, the expected cost of Step 2 is $O(n^2 \log n \log\log n)$, by Lemma 9, which also bounds the complexity of Step 3 (assuming it is reached), since we actually expect to discover that p is composite using just $O(1)$ Miller–Rabin tests. The probability of reaching Step 4 is less than $4^{-\log x} = O(1/x)$, by [22], which makes the conditional cost of Steps 4 and 5 in this case completely negligible since they both have expected running times that are polynomial in $\log x$.
2. If p is prime and N is composite, then the expected cost of Step 2 given by Lemma 9 is $O(n^4(\log n)^2 \log\log n)$, which dominates the complexity of Step 3 and the conditional cost of Step 4 (which, as in Case 1, we have a negligible probability of reaching).
3. If p and N are prime, then the expected costs of Steps 2–5 are, respectively, $O(n^4(\log n)^2 \log\log n)$, $O(n^3 \log n \log\log n)$, $O(n^{4+\varepsilon})$, and $O(n^2 \log n \log\log n)$; see [3] for the bound on Step 4. Thus, the total expected cost is $O(n^{4+\varepsilon})$ for any $\varepsilon > 0$.

We now bound the expected running time of Algorithm 2 by considering how often we expect each case to occur. We expect to be in Case 1 for $O(n^2 \log n)$ triples, each of which takes $O(n^2 \log n \log\log n)$ expected time, yielding a total bound of $O(n^{4+\varepsilon})$. We expect to be in Case 2 for $O(n \log n)$ triples, each of which takes $O(n^4(\log n)^2 \log\log n)$ expected time, yielding a total bound of $O(n^5(\log n)^3 \log\log n)$. Case 3 occurs exactly once and takes $O(n^{4+\varepsilon})$ expected time. Case 2 dominates and the theorem follows.

6 Comments

The bound in Theorem 3 would be improved by a factor of $\log n$ if one could show that $H(D) = \Omega(\sqrt{|D|})$, on average. But this appears to be beyond our present capabilities. First, we note that the distribution of D is not uniform, and even in the case of uniformly chosen discriminants D, the average value of $H(D)$ is not known; see [12, Sect. 15.9] for a discussion.

As a practical optimization, one can add an *early abort* option in Algorithm 1 that causes the algorithm to terminate if it discovers that $N \equiv 0 \bmod \ell$. Heuristically, this should reduce the running time of Algorithm 2 by a factor of $\log n$. Another practical optimization is to reuse the modular polynomials Φ_ℓ that are computed in Algorithm 1, which do not depend on the inputs p, a, and b. This saves a factor of $\log n$ in the expected running time but increases the expected space complexity from $O(n^3 \log n)$ to $O(n^4 \log n)$. Combining these two optimizations with the assumption that $H(D) \gg \sqrt{|D|}$ on average yields a heuristic expected running time of $O(n^5 \log\log n)$ for Algorithm 2.

Acknowledgments During the preparation I. E. Shparlinski was supported in part by ARC Grant DP130100237 and by NRF Grant CRP2-2007-03, Singapore. A. V. Sutherland received financial support from NSF Grant DMS-1115455.

References

1. M. Agrawal, N. Kayal, and N. Saxena, 'PRIMES is in P', *Ann. Math.,* **160** (2004), 781–793.
2. R. Avanzi, H. Cohen, C. Doche, G. Frey, T. Lange, K. Nguyen and F. Vercauteren, *Elliptic and hyperelliptic curve cryptography: Theory and practice*, CRC Press, 2005.
3. D. Bernstein, 'Proving primality in essentially quartic random time', *Math. Comp.,* **76** (2007), 389–403.
4. A. Bostan, B. Salvy, F. Morain and É. Schost, 'Fast algorithms for computing isogenies between elliptic curves', *Math. Comp.,* **77** (2008), 1755–1778.
5. R. Bröker, K. Lauter and A. V. Sutherland, 'Modular polynomials via isogeny volcanoes', *Math. Comp.,* posted on July 14, 2011, PII S 0025–5718(2011) 02508–1, to appear in print.
6. D. A. Cox, *Primes of the form* $x^2 + ny^2$*: Fermat, class field theory, and complex multiplication*, John Wiley & Sons, New York, 1989.
7. N. D. Elkies, 'Elliptic and modular curves over finite fields and related computational issues', *Computational perspectives on number theory*, D. A. Buell and J. T. Teitelbaum eds., *Studies in Advanced Mathematics*, Amer. Math. Soc., Providence, RI, **7** (1998), 21–76.
8. S. Galbraith, *Mathematics of public key cryptography*, Cambridge University Press, 2012.
9. J. von zur Gathen and J. Gerhard, *Modern computer algebra*, 2nd ed., Cambridge University Press, 2003.
10. P. Gaudry and F. Morain, 'Fast algorithms for computing the eigenvalue in the Schoof-Elkies-Atkin algorithm', *Intern. Symp. on Symbolic and Algebraic Comput. (ISSAC 2006)*, ACM, 2006, 109–115.
11. D. Hankerson, A. Menezes, and S. Vanstone, *Guide to ellitpic curve cryptography*, Springer, New York, 2004.
12. H. Iwaniec and E. Kowalski, *Analytic number theory*, Amer. Math. Soc., Providence, RI, 2004.
13. E. Jeong, 'Isomorphism classes of elliptic curves over finite fields with characteristic 3', *J. Chungcheong Math. Soc.*, 22 (2009), 207–213.
14. N. Koblitz, 'Elliptic curve implementation of zero-knowledge blobs', *J. Cryptology*, 4 (1991), 207–213.
15. H. W. Lenstra, 'Factoring integers with elliptic curves', *Ann. Math.*, **126** (1987), 649–673.
16. J. E. Littlewood, 'On the class-number of the corpus $P(\sqrt{-k})$', *Proc. London Math. Soc.*, **27** (1928), 358–372.
17. R. Lidl and H. Niederreiter, *Finite fields*, Cambridge Univ. Press, Cambridge, 1997.
18. M. Lochter and J. Merkle, *Elliptic curve cryptography (ECC) brainpool standard curves and curve generataion*, https://tools.ietf.org/html/rfc5639, 2010.
19. F. Luca and I. E. Shparlinski, 'On quadratic fields generated by polynomials', *Arch. Math.*, **91** (2008), 399–408.
20. J. McKee, 'Subtleties in the distribution of the numbers of points on elliptic curves over a finite prime field', *J. London Math. Soc.*, **59** (1999), 448–460.
21. H. L. Montgomery and R. C. Vaughan, *Multiplicative number theory I: Classical theory*, Cambridge Univ. Press, Cambridge, 2006.
22. M. Rabin, 'Probabilistic algorithms for testing primality', *J. Number Theory*, **12** (1980), 128–138.
23. T. Satoh, 'On p-adic point counting algorithms for elliptic curves over finite fields', *Lect. Notes in Comp. Sci.*, Springer-Verlag, Berlin, **2369** (2002), 43–66.
24. A. Schönhage and V. Strassen, 'Schnelle Multiplikation großer Zahlen', *Computing*, **7** (1971), 281–292.
25. R. Schoof, 'Counting points on elliptic curves over finite fields', *J. Théorie des Nombres de Bordeaux*, **7** (1995), 219–254.
26. I. E. Shparlinski, 'On the product of small Elkies primes', *Proc. Amer. Math. Soc.* (to appear).
27. J. H. Silverman, *The arithmetic of elliptic curves*, 2nd ed., Springer, Dordrecht, 2009.
28. A. V. Sutherland, 'Identifying supersingular elliptic curves', *LMS J. Comp. and Math.*, **15** (2012), 317–325.
29. A. V. Sutherland, 'Computing Hilbert class polynomials with the Chinese Remainder Theorem', *Math. Comp.*, **80** (2011), 501–538.