Draft, July 6, 2011
Volume 00, Number 0, Pages 000–000
S (XX)0000-0

IDENTIFYING SUPERSINGULAR ELLIPTIC CURVES

ANDREW V. SUTHERLAND

Abstract. Given an elliptic curve E over a field of characteristic p, we consider how to efficiently determine whether E is ordinary or supersingular. We analyze the complexity of several existing algorithms and then present a new approach that exploits structural differences between ordinary and supersingular isogeny graphs. This yields a simple algorithm that, given E and a suitable non-residue in $\mathbb{F}_{p^2}$, determines the supersingularity of E in $O(n^3 \log^2 n)$ time and $O(n)$ space, where $n = O(\log p)$. Both these complexity bounds are significant improvements over existing methods, as we demonstrate with some practical computations.

1. Introduction

An elliptic curve E over a field F of prime characteristic p is called *supersingular* if its p-torsion subgroup $E[p](\bar{F})$ is trivial; see [7, §13.7] or [19, §V.3] for several equivalent definitions. Otherwise, we say that E is *ordinary*. Supersingular curves differ from ordinary curves in many ways, and this has practical implications for algorithms that work with elliptic curves over finite fields, such as algorithms for counting points [16], generating codes [17], computing endomorphism rings [8], and calculating discrete logarithms [10]. Given an elliptic curve, one of the first things we might wish to know is whether it is ordinary or supersingular, and we would like to make this distinction as efficiently as possible.

The answer to this question depends only on the isomorphism class of E over $\bar{F}$, which is characterized by its j-invariant $j(E)$. It is known that E can be supersingular only when $j(E) \in \mathbb{F}_{p^2}$, thus we may restrict our attention to the case that F is a finite field $\mathbb{F}_q \subseteq \mathbb{F}_{p^2}$. We also recall that E is supersingular if and only if $\#E(\mathbb{F}_q) \equiv 1 \bmod p$; see [19] for proofs of these facts.

There is a simple Monte Carlo test that quickly identifies ordinary elliptic curves. When $q = p$, one picks a random point P on the curve and computes the scalar multiple $(p+1)P$. If $(p+1)P \neq 0$ then the curve is ordinary, and if $(p+1)P = 0$ then the curve is likely to be supersingular (see §2.3 for the case $q = p^2$). If several repetitions of this test fail to prove that E is ordinary, then it is almost certainly supersingular. But this approach cannot *prove* that E is supersingular, just as the Miller-Rabin primality test [11] cannot prove that an integer is prime.

To prove that E is supersingular, one may verify that $\#E(\mathbb{F}_q) \equiv 1 \bmod p$ using a point-counting algorithm, such as Schoof's algorithm [15, 16]. With a variant of the SEA algorithm (see §2.2), this can be accomplished in $O(n^4 \operatorname{llog} n)$ time using $O(n^4)$ space, where $n = \log q$. The computer algebra systems Magma [2] and Sage [20] both use this approach to identify supersingular curves.

2010 *Mathematics Subject Classification.* Primary 11G07 ; Secondary 11Y16, 11G20, 14H52.

But it is natural to ask whether one can do better. We show that this is indeed the case, presenting an algorithm that runs in $O(n^3 \log^2 n)$ time and $O(n)$ space. Rather than counting points, we rely on structural differences between ordinary and supersingular isogeny graphs. The resulting algorithm is easy to implement and much faster than methods based on point-counting (as may be seen in Table 1).

In the first step of the algorithm we must solve a cubic equation, and in each subsequent step we need to solve a quadratic equation. To obtain a deterministic algorithm, we assume that we are given a quadratic non-residue and a cubic non-residue in $\mathbb{F}_{p^2}$ to facilitate these computations. When $\mathbb{F}_{p^2}$ is constructed using a generator, this generator already provides the non-residues we require. Alternatively, non-residues can be efficiently obtained by sampling random elements, yielding a Las Vegas algorithm.

2. Existing Algorithms

Before presenting the new algorithm, we briefly review some standard methods for testing supersingularity and analyze their complexity. Over fields of characteristic 2 or 3, an elliptic curve E is supersingular if and only if $j(E) = 0$, a condition that is trivial to check given an equation for the curve. As noted in the introduction, we may assume E is defined over $\mathbb{F}_{p^2}$ (otherwise E is ordinary). Thus we shall work over a finite field $\mathbb{F}_q$ of characteristic $p > 3$, where q is either p or p^2.

We use $\mathsf{M}(n)$ to denote the cost of multiplying two n-bit integers, which we may bound by $\mathsf{M}(n) = O(n \log n \operatorname{llog} n) = \tilde{O}(n)$, via [14]. All of our bounds are expressed in terms of $n = \log p$, which is proportional to the size of the input for our problem, the coefficients of the curve E.

2.1. **Exponential time algorithms.** If E is in Weierstrass form $y^2 = f(x)$, then E is supersingular if and only if the coefficient of x^{p-1} in $f(x)^{(p-1)/2}$ is zero, and this implies that if E is in Legendre form $y^2 = x(x-1)(x-\lambda)$, then E is supersingular if and only if $\sum_{i=0}^{m} \binom{m}{i}^2 \lambda^i = 0$; see [19, Thm. 4.1]. These criterion are convenient and easy to state, but they are computationally useful only when p is very small, since the time required to apply them is exponential in n.

2.2. **Polynomial time algorithms.** Schoof's algorithm [15] computes $\#E(\mathbb{F}_q)$ in $\tilde{O}(n^5)$ time and $O(n^3)$ space. This immediately yields a deterministic polynomial-time algorithm for testing supersingularity, since E is supersingular if and only if $\#E(\mathbb{F}_q) \equiv 1 \bmod p$. The improvements of Elkies and Atkin incorporated in the SEA algorithm [4, 16] are not immediately applicable, since they rely on results that do not necessarily apply to supersingular curves [16, Prop. 6.1-3]. However, as remarked by Schoof [16, p. 241], supersingular curves can be identified using similar techniques. Let us briefly fill in the details.

Recall that for any prime $\ell \neq p$, the classical modular polynomial $\Phi_\ell \in \mathbb{Z}[X, Y]$ has the property that two j-invariants $j_1, j_2 \in \mathbb{F}_q$ satisfy $\Phi_\ell(j_1, j_2) = 0$ if and only if $j_1 = j(E_1)$ and $j_2 = j(E_2)$ for some elliptic curves E_1 and E_2 related by a cyclic isogeny of degree ℓ; see [9, Thm. 12.19]. If E_1 and E_2 are isogenous, then $\#E_1(\mathbb{F}_q) = \#E_2(\mathbb{F}_q)$, thus E_1 is supersingular if and only if E_2 is. Since every supersingular j-invariant in characteristic p lies in $\mathbb{F}_{p^2}$, if E is supersingular then the univariate polynomial $\phi_{\ell,E}(X) = \Phi_\ell(j(E), X)$ splits completely in $\mathbb{F}_{p^2}[X]$, for every prime $\ell \neq p$. However, if E is ordinary, this is not the case.

Proposition 1. *Let $j(E) \in \mathbb{F}_{p^2}$ and assume $j(E) \neq 0, 1728$.*[1] *Let S be a set of primes $\ell \neq p$ with product $M > 2p$. Then E is supersingular if and only if $\phi_{\ell,E}$ splits completely in $\mathbb{F}_{p^2}[X]$ for every $\ell \in S$.*

Proof. The forward implication is addressed in the discussion above. For the reverse, suppose for the sake of contradiction that E is ordinary and that $\phi_{\ell,E}$ splits completely in $\mathbb{F}_{p^2}[X]$ for all $\ell \in S$. It follows from [5, Thm. 2.1] (or see §3) that $t^2 - 4p^2$ is divisible by ℓ^2, where $t = p^2 + 1 - \#E(\mathbb{F}_{p^2})$ is the trace of Frobenius of $E/\mathbb{F}_{p^2}$. Thus $t^2 \equiv 4p^2 \bmod \ell^2$ for each $\ell \in S$, and therefore $t^2 \equiv 4p^2 \bmod M^2$, by the Chinese Remainder Theorem. The Hasse bound implies $t^2 \leq 4p^2$, so we must have $t^2 = 4p^2$, since $M^2 > 4p^2$. Thus $t = \pm 2p$, and therefore $\#E(\mathbb{F}_{p^2}) \equiv 1 \bmod p$. But this implies that E is supersingular, which is a contradiction. □

To prove the supersingularity of $E/\mathbb{F}_q$, it suffices to verify that $\phi_{\ell,E}$ splits completely in $\mathbb{F}_{p^2}[X]$ for each of the first m primes ℓ with product $M > 2p$. This can be done without factoring $\phi_{\ell,E}$. One removes all linear factors from $\phi_{\ell,E}$ as follows: first let $f = \phi_{\ell,E}$ and compute $g = \gcd(f(X), X^p - X)$, then repeatedly set $f \leftarrow f/g$ and $g \leftarrow \gcd(f, g)$ until $\deg g = 0$. If at this point $\deg f = 0$, then $\phi_{\ell,E}$ splits completely over $\mathbb{F}_{p^2}$ and otherwise it does not. When $j(E)$ lies in $\mathbb{F}_p$, we may instead work in $\mathbb{F}_p[X]$ and remove both linear and quadratic factors from $\phi_{\ell,E}$ with a similar approach.

Using precomputed modular polynomials, this yields a deterministic algorithm that runs in $O(n^2\mathsf{M}(n^2)/\log n) = O(n^4 \operatorname{llog} n)$ time and $O(n^4)$ space, assuming Kronecker substitution [24, §8.4] is used to multiply polynomials in $\mathbb{F}_{p^2}[X]$ of degree $O(n)$ in time $O(\mathsf{M}(n^2))$. The space can be reduced to $O(n^3 \log n)$ by computing modular polynomials as required, but this significantly increases the running time.

2.3. **A Monte Carlo algorithm.** For a supersingular curve E over a finite field of characteristic $p > 3$ it follows from [13] that

(i) if E is defined over $\mathbb{F}_p$ then $\#E(\mathbb{F}_p) = p + 1$;
(ii) either $E(\mathbb{F}_{p^2}) \cong (\mathbb{Z}/(p-1)\mathbb{Z})^2$ or $E(\mathbb{F}_{p^2}) \cong (\mathbb{Z}/(p+1)\mathbb{Z})^2$.

This motivates the following algorithm.

Algorithm 1. *Given an elliptic curve $E/\mathbb{F}_q$ with $q|p^2$:*

1. **If $q = p$:** pick a random point $P \in E(\mathbb{F}_p)$ and return **true** if $(p + 1)P = 0$, otherwise return **false**.
2. **If $q = p^2$:** pick a random point $P \in E(\mathbb{F}_{p^2})$ and return **true** if either $(p - 1)P = 0$ or $(p + 1)P = 0$, otherwise return **false**.

If the algorithm returns **false** then E is ordinary. We now show that if the algorithm returns **true**, then E is very likely to be supersingular (for large q).

Proposition 2. *Given an ordinary elliptic curve $E/\mathbb{F}_q$, Algorithm 1 returns* **true** *with probability at most $8\sqrt{q}/(\sqrt{q} - 1)^2 = O(q^{-1/2})$.*

Proof. First, let $q = p$. Let H be the $(p + 1)$-torsion subgroup $E(\mathbb{F}_q)[p + 1]$. Then $H \cong \mathbb{Z}/m_1\mathbb{Z} \times \mathbb{Z}/m_2\mathbb{Z}$, where m_1 divides m_2 and $q - 1$. Since m_1 also divides $p + 1$, we have $m_1 \leq 2$. We now show $m_2 \leq 4\sqrt{q}$. If not, then $p + 1$ is the unique multiple of m_2 in the Hasse interval $[(\sqrt{q} - 1)^2, (\sqrt{q} + 1)^2]$. But then $\#E(\mathbb{F}_p) = p + 1$, contradicting the fact that E is ordinary. Thus $\#H = m_1 m_2 \leq 8\sqrt{q}$.

---

[1] We note that $j(E) = 0$ (resp. 1728) is supersingular if and only if $p \not\equiv 1 \bmod 3$ (resp. 4).

Now let $q = p^2$. Let H be the union of $H_1 = E(\mathbb{F}_q)[p-1]$ and $H_2 = E(\mathbb{F}_q)[p+1]$. Then $\#H_1 \le 4\sqrt{q}$, else $(p-1)^2$ is the unique multiple of $\#H_1$ in the Hasse interval, yielding a contradiction as above. Similarly, $\#H_2 \le 4\sqrt{q}$, and therefore $\#H \le 8\sqrt{q}$.

In both cases, Algorithm 1 outputs **true** only when the random point P lies in H, which occurs with probability $\#H/\#E(\mathbb{F}_q) \le 8\sqrt{q}/(\sqrt{q}-1)^2$. □

Algorithm 1 is a Monte Carlo algorithm with one-sided error. For $q \ge 7$ the error probability given by Proposition 2 is bounded below 1 and can be made arbitrarily small (but never zero) by repetition. Using standard techniques, the random point[2] P can be obtained in $O(n\mathsf{M}(n)) = \tilde{O}(n^2)$ expected time, and this also bounds the cost of the scalar multiplications.

3. Isogeny graphs

As above, we work in a finite field $\mathbb{F}_q$ of characteristic $p > 3$. For each prime $\ell \ne p$ we define the (directed multi-) graph $G_\ell(\mathbb{F}_q)$ of $\mathbb{F}_q$-rational ℓ-isogenies.

Definition 1. *$G_\ell(\mathbb{F}_q)$ is the graph with vertex set $\mathbb{F}_q$ and edges (j_1, j_2) present with multiplicity k whenever j_2 is a root of $\Phi_\ell(j_1, X)$ with multiplicity k.*

As in §2.2, the polynomial $\Phi_\ell \in \mathbb{Z}[X,Y]$ is the classical modular polynomial that parametrizes ℓ-isogenous pairs of j-invariants; see [9, §5.2]. It is symmetric and has degree $\ell + 1$ in both variables, thus the in-degree and out-degree of each vertex of $G_\ell(\mathbb{F}_q)$ is at most $\ell + 1$. These degrees need not coincide (e.g., for the vertices 0, 1728, and their neighbors); when we speak of the *degree* of a vertex we refer to its out-degree. We note that $G_\ell(\mathbb{F}_q)$ may contain self-loops, edges of the form (j_1, j_1).

Each vertex of $G_\ell(\mathbb{F}_q)$ is the j-invariant $j(E)$ of an elliptic curve E defined over $\mathbb{F}_q$, and we may classify each vertex as ordinary or supersingular. We may similarly classify the edges and connected components of $G_\ell(\mathbb{F}_q)$, since every edge lies between vertices of the same type (ordinary or supersingular). As noted in §2.2, if $j(E)$ is a supersingular j-invariant then the polynomial $\Phi_l(j(E), X)$ splits completely in $\mathbb{F}_{p^2}[X]$, and it follows that for $q > p$, every supersingular component[3] of $G_\ell(\mathbb{F}_q)$ is a regular graph of degree $\ell + 1$.

However, the ordinary components of $G_\ell(\mathbb{F}_q)$ are *not* regular graphs of degree $\ell + 1$; they contain many vertices of degree less than $\ell + 1$, and this is the basis of our algorithm. Given an elliptic curve E defined over $\mathbb{F}_{p^2}$, our strategy is to search for a vertex of degree less than 3 that is connected to $j(E)$ in $G_2(\mathbb{F}_{p^2})$. If we find such a vertex, then E is ordinary, and if we can prove no such vertex exists, then E is supersingular. To do this we need to understand the structure of the ordinary components of $G_2(\mathbb{F}_{p^2})$. All the facts we require apply more generally to $G_\ell(\mathbb{F}_q)$, so we continue in this setting.

A detailed analysis of the structure of the ordinary components of $G_\ell(\mathbb{F}_q)$ was undertaken by Kohel in his thesis [8], and they are now commonly called *ℓ-volcanoes*, a term introduced by Fouquet and Morain [5]. The structure of an ℓ-volcano is determined by the relationships between the endomorphism rings of the elliptic curves corresponding to its vertices. Here we record only the facts we need, referring to [5, 8] for proofs and a more complete presentation.

---

[2] In practice one does not use a uniform distribution over $E(\mathbb{F}_q)$, one constructs a uniformly random point in $E(\mathbb{F}_q) - E(\mathbb{F}_q)[2]$. This is easier, and better, for the purposes of the algorithm.

[3] There is in fact only one supersingular component of $G_\ell(\mathbb{F}_{p^2})$; see [8, Cor. 78].

Let $j(E)$ be a vertex in an ordinary component V of $G_\ell(\mathbb{F}_q)$ (an ℓ-volcano). Recall that the endomorphism ring of an ordinary elliptic curve is isomorphic to an order $\emptyset$ in an imaginary quadratic field K. We have the inclusions $\mathbb{Z}[\pi] \subseteq \emptyset \subseteq \emptyset_K$, where $\mathbb{Z}[\pi]$ is the order generated by (the image of) the Frobenius endomorphism π, and $\emptyset_K$ is the maximal order of K (its ring of integers). The order $\emptyset$ depends only on the isomorphism class $j(E)$, while the orders $\mathbb{Z}[\pi]$ and O_K depend only on the isogeny class of E and are invariants of V.

We may partition the vertices of V into *levels* $V_0, \ldots, V_d$, where the level V_i in which $j(E)$ lies is determined by the ℓ-adic valuation $i = \nu_\ell([\emptyset_K : \emptyset])$. The integer $d = \nu_\ell[\emptyset_K : \mathbb{Z}[\pi]]$ is the *depth* (also called the *height*) of V, and may be 0. From the norm equation

$$4q = t^2 - v^2 D, \tag{1}$$

where $q = N(\pi)$, $t = \operatorname{tr} \pi$, $D = \operatorname{disc}(K)$, and $d = \nu_\ell(v)$, we have

$$d < \log_\ell \sqrt{4q}. \tag{2}$$

Level V_d is the *floor* of the ℓ-volcano V. Its vertices are distinguished by their degree, which is at most 2. Every other vertex in V (if any) has degree $\ell + 1$.

Proposition 3. *Let $j(E)$ be a vertex at level V_i of an ℓ-volcano V of depth d.*

(i) *The degree of $j(E)$ is $\ell + 1$ if and only if $i < d$.*
(ii) *If $i = 0 < d$ then at least $\ell - 1$ of the edges from $j(E)$ lead to V_1.*
(iii) *If $0 < i < d$ then one edge from $j(E)$ leads to V_{i-1} and the rest lead to V_{i+1}.*
(iv) *If $0 < i = d$ then $j(E)$ has just one outgoing edge and it leads to V_{d-1}.*

Proof. See [5, Thm. 2.1] and [8, Prop. 23]. □

Given $E/\mathbb{F}_q$, our goal is to either find a path from $j(E)$ to the floor of its ℓ-volcano in $G_\ell(\mathbb{F}_q)$, or prove that no such path exists. We define a path as follows.

Definition 2. *A path (of length k) in $G_\ell(\mathbb{F}_q)$ is a sequence of vertices $j_0, j_1, \ldots, j_k$ such that $\Phi_\ell(j_0, j_1) = 0$ and j_{i+2} is a root of $\Phi_\ell(j_{i+1}, X)/(X - j_i)$ for $0 \le i < k-1$.*

In terms of a walk on the graph, this definition prohibits backtracking except when there are multiple edges leading back to the previous vertex. Edges that lead toward the floor (from level V_i to V_{i+1}) are called *descending*. Proposition 3 implies that every vertex of V not on the floor has at least $\ell - 1$ descending edges. Any path that starts with a descending edge can only be extended by descending further, and this must lead to the floor within d steps (this is called a *descending path* in [5]).

We can summarize these results in purely graph-theoretic terms. For any edge (j_0, j_1) in $G_\ell(\mathbb{F}_q)$, not necessarily ordinary, let $R_k(j_0, j_1)$ denote the set of vertices j_k for which there exists a path $j_0, j_1, \ldots, j_k$ of length k.

Corollary 1. *Let j_0 be a vertex of $G_\ell(\mathbb{F}_q)$ of degree $\ell + 1$.*

(i) *If j_0 is ordinary, then $G_\ell(\mathbb{F}_q)$ contains $\ell - 1$ edges (j_0, j_1) for which the set $R_k(j_0, j_1)$ is empty for some $1 \le k < \log_\ell \sqrt{4q} + 1$.*
(ii) *If j_0 is supersingular and $q > p$, then for every edge (j_0, j_1) the set $R_k(j_0, j_1)$ is nonempty for all $k \ge 1$.*

4. The algorithm

We now present our algorithm, which, given an elliptic curve over a field of positive characteristic, returns **true** if E is supersingular and **false** otherwise.

Algorithm 2. *Given an elliptic curve E/F with $\operatorname{char} F = p > 0$:*

1. If $j(E) \notin \mathbb{F}_{p^2}$ then return **false**.
2. If $p \leq 3$ then return **true** if $j(E) = 0$ and **false** otherwise.
3. Attempt to find three roots j_1, j_2, j_3 of $\Phi_2(j(E), X)$ in $\mathbb{F}_{p^2}$.
   If $\Phi_2(j(E), X)$ does not have three roots in $\mathbb{F}_{p^2}$ then return **false**.
4. Set $j_i' \leftarrow j(E)$ for $i = 1, 2, 3$.
5. Let $m = \lfloor \log_2 p \rfloor + 1$, and for $k = 1$ to m:
   a. Set $f_i(X) \leftarrow \Phi_2(j_i, X)/(X - j_i')$ and set $j_i' \leftarrow j_i$, for $i = 1, 2, 3$.
   b. Attempt to find a root j_i of $f_i(X)$ in $\mathbb{F}_{p^2}$, for $i = 1, 2, 3$.
      If any $f_i(X)$ does not have a root in $\mathbb{F}_{p^2}$ then return **false**.
6. Return **true**.

After ruling out some trivial cases, the algorithm begins in step 3 by computing the outgoing edges from the vertex $j(E)$ in $G_2(\mathbb{F}_{p^2})$, using the modular polynomial

$$\begin{aligned}\Phi_2(X, Y) = X^3 + Y^3 - X^2Y^2 + 1488(X^2Y + Y^2X) - 162000(X^2 + Y^2)\\ + 40773375XY + 8748000000(X + Y) - 157464000000000.\end{aligned}$$

If the vertex $j(E)$ does not have degree 3 then E must be ordinary and the algorithm terminates. Otherwise, it attempts to extend each of the three edges $(j(E), j_i)$ to a path of length $m + 1 > \log_2 \sqrt{4p^2} + 1$ in step 5. If E is ordinary than one of these attempts must fail, and otherwise E must be supersingular, by Corollary 1.

Thus the algorithm is correct. We now analyze its complexity, considering two possible implementations, one probabilistic and one deterministic. As in §2, we let $\mathsf{M}(n)$ denote the cost of multiplication and express our bounds in terms of $n = \log p$.

4.1. Probabilistic complexity analysis.

The work of Algorithm 2 consists essentially of solving a cubic equation in step 3 and at most $3m = O(n)$ quadratic equations in step 5. With a probabilistic root-finding algorithm [24, Alg 14.5], we expect to use $O(n)$ operations in $\mathbb{F}_{p^2}$ for each equation, yielding a total expected running time of $O(n^2)$ operations in $\mathbb{F}_{p^2}$, using storage for $O(1)$ elements of $\mathbb{F}_{p^2}$. This gives an expected running time of $O(n^2\mathsf{M}(n)) = O(n^3 \log n \operatorname{llog} n)$ using $O(n)$ space. The output of the algorithm is not affected by any of the random choices that are made (it is always correct), thus we have a Las Vegas algorithm.

Proposition 4. *Algorithm 2 can be implemented as a Las Vegas algorithm with an expected running time of $O(n^3 \log n \operatorname{llog} n)$ using $O(n)$ space.*

4.2. Deterministic complexity analysis.

We now consider how we may obtain a deterministic algorithm, given some additional information. First, we note that the choice of the root j_i in step 5 can be fixed by ordering $\mathbb{F}_{p^2}$ with respect to some basis. Second, we may apply the quadratic formula and Cardano's method (valid over any field of characteristic not 2 or 3), to solve the equations arising in steps 3 and 5 by radicals. To find the roots of a quadratic or cubic polynomial that splits completely in $\mathbb{F}_{p^2}[X]$, it suffices to compute square roots and cube roots in $\mathbb{F}_{p^2}$.

For any prime r, computing an rth root in a finite field $\mathbb{F}_q$ can be reduced to an exponentiation and a (possibly trivial) discrete logarithm computation in the r-Sylow subgroup of $\mathbb{F}_q^*$. For $r = 2$ this is the Tonelli-Shanks algorithm [23, 18], and the generalization to $r > 2$ is due to Adleman, Manders, and Miller [1]. For the discrete logarithm computation we require a generator γ for the r-Sylow subgroup H_r of $\mathbb{F}_q^*$ (which is necessarily cyclic). Using the algorithm in [22] we can compute discrete logarithms in H_r using $O(n \log n / \operatorname{llog} n)$ operations in $\mathbb{F}_q$, assuming r and the degree of $\mathbb{F}_q$ are fixed. This yields a bit-complexity of $O(\mathsf{M}(n) n \log n / \operatorname{llog} n) = O(n^2 \log^2 n)$, which dominates the cost of exponentiation.

When H_r is not trivial, any element α of $\mathbb{F}_q$ that is not an rth power residue yields a generator for H_r: simply let $\gamma = \alpha^{(q-1)/s}$, where $s = r^{\nu_r(q-1)}$. This yields the following proposition.

Proposition 5. *Algorithm 2 can be implemented as a deterministic algorithm that runs in $O(n^3 \log^2 n)$ time using $O(n)$ space, given a quadratic non-residue and a cubic non-residue in $\mathbb{F}_{p^2}$.*

As noted earlier, we can efficiently obtain non-residues by sampling random elements. Given a uniformly random $\alpha \in \mathbb{F}_q^*$, if we let $\gamma = \alpha^{(q-1)/s}$ as above, then γ generates H_r if and only if $\gamma^{s/r} \neq 1$, which occurs with probability $(r-1)/r$. Alternatively, if we are given a generator for $\mathbb{F}_{p^2}$ (the coefficients of E may be specified in terms of such a generator), then we already have an element that is both a quadratic and a cubic non-residue.

We remark that while the complexity bound in Proposition 5 is slightly worse than the bound in Proposition 4, in practice the deterministic approach is usually faster because the 2-Sylow and 3-Sylow subgroups of most finite fields are very small, meaning that the discrete logarithm computations take negligible time.

4.3. **Average case complexity.** The bounds given in Propositions 4 and 5 are worst-case complexity bounds. We now consider the performance of Algorithm 2, on average, when given a random elliptic curve over $\mathbb{F}_{p^2}$.

Proposition 6. *Given an elliptic curve whose j-invariant is uniformly distributed over $\mathbb{F}_{p^2}$, the expected running time of Algorithm 2 is $O(n^2 \log n \operatorname{llog} n)$.*

Proof. By [19, Thm. 4.1], the proportion of supersingular j-invariants in $\mathbb{F}_{p^2}$ is $O(1/p)$. It follows from Propositions 4 and 5 that these cases have a negligible impact on the expected running time. Given an ordinary elliptic curve with j-invariant j_0, the running time of Algorithm 2 is $O(n\mathbf{E}[d - i + 1])$ field operations, where d is the depth of the 2-volcano in $G_2(\mathbb{F}_{p^2})$ containing j_0, and V_i is the level in which j_0 lies. By Proposition 3, for $d > 0$ we have $\#V_0 \leq \#V_1$ and $\#V_i = \#V_{i+1}/2$, for $0 < i < d$. This implies that $\mathbf{E}[d-i+1]$ is $O(1)$, and the proposition follows. □

The bound in Proposition 6 applies to both the probabilistic and deterministic implementations of Algorithm 2 considered above. With a probabilistic implementation, the expected running time of Algorithm 2 is within a constant factor of the running time of the Monte Carlo approach used in Algorithm 1, and for almost all values of p (those for which $p^2 - 1$ is not divisible by an unusually large power of 2 or 3), this is also true of the deterministic implementation. Remarkably, this constant factor actually favors Algorithm 2, which identifies most ordinary curves even more quickly than Algorithm 1 (as may be seen in Table 1).

TABLE 1. Performance results (CPU times in milliseconds).

	ordinary				supersingular			
	Magma		Alg. 2		Magma		Alg. 2	
b	$\mathbb{F}_p$	$\mathbb{F}_{p^2}$	$\mathbb{F}_p$	$\mathbb{F}_{p^2}$	$\mathbb{F}_p$	$\mathbb{F}_{p^2}$	$\mathbb{F}_p$	$\mathbb{F}_{p^2}$
64	1	25	0.1	0.1	226	770	2	8
128	2	60	0.1	0.1	2010	9950	5	13
192	4	99	0.2	0.1	8060	41800	8	33
256	7	140	0.3	0.2	21700	148000	20	63
320	10	186	0.4	0.3	41500	313000	39	113
384	14	255	0.6	0.4	95300	531000	66	198
448	19	316	0.8	0.5	152000	789000	105	310
512	24	402	1.0	0.7	316000	2280000	164	488
576	30	484	1.3	0.9	447000	3350000	229	688
640	37	595	1.6	1.0	644000	4790000	316	945
704	46	706	2.0	1.2	847000	6330000	444	1330
768	55	790	2.4	1.5	1370000	8340000	591	1770
832	66	924	3.1	1.9	1850000	10300000	793	2410
896	78	1010	3.2	2.1	2420000	12600000	1010	3040
960	87	1180	4.0	2.5	3010000	16000000	1280	3820
1024	101	1400	4.8	3.1	5110000	35600000	1610	4880

5. Computational results

Table 1 compares the performance of Algorithm 2 with the implementation of the IsSupersingular function provided by the Magma computer algebra system. The Magma implementation relies on two standard methods for distinguishing supersingular curves: it first performs a Monte Carlo test to quickly identify ordinary curves (as in Algorithm 1), and then applies the modular polynomial approach described in §2.2. Our implementation was built on the Gnu Multiple Precision Arithmetic Library (GMP) [6], which is also used by Magma. All tests were run on a single core of an AMD Opteron 250 processor clocked at 2.4 GHz.

Each row of Table 1 corresponds to a series of tests using a fixed bit-length b. For each value of b we selected 5 random primes p in the interval $[2^{b-1}, 2^b]$, and for each prime p we generated 100 elliptic curves defined over $\mathbb{F}_p$ and 100 elliptic curves defined over $\mathbb{F}_{p^2}$, with uniformly distributed j-invariants. As one might expect, all of these randomly generated curves were ordinary, and the average times to process these curves are listed in the "ordinary" columns of Table 1.

To test performance on supersingular inputs, for each prime p we constructed a supersingular curve over $\mathbb{F}_p$ using a variant of the CM method described in [3]. This involves picking a discriminant $D < 0$ with $\left(\frac{D}{p}\right) = -1$ and $-D$ prime. The Hilbert class polynomial $H_D(X)$ is then guaranteed to have an $\mathbb{F}_p$-rational root j_0, which is necessarily the j-invariant of a supersingular elliptic curve. In order for this to be feasible, the discriminant D cannot be too large; we used random discriminants in the interval $[2^{31}, 2^{32}]$, and computed $H_D(X) \bmod p$ using the algorithm in [21].

Over $\mathbb{F}_p$, the supersingular j-invariants obtained in this fashion are not uniformly distributed over the set of all supersingular j-invariants in $\mathbb{F}_p$. However, one expects the running times of both Algorithm 2 and the Magma implementation to be essentially independent of D, and this appears to be the case. Over $\mathbb{F}_{p^2}$, we are able to obtain a nearly uniform distribution of supersingular j-invariants by performing a random walk on the graph $G_2(\mathbb{F}_{p^2})$, starting from a vertex defined over $\mathbb{F}_p$ constructed using the CM method described above. The supersingular component S of $G_2(\mathbb{F}_{p^2})$ is a Ramanujan graph [12], and this implies that, starting from any vertex of S, a random walk of $O(n)$ steps on S yields a nearly uniform distribution on its vertices.

5.1. **Discussion of results.** Table 1 clearly demonstrates a significant performance advantage for Algorithm 2, both asymptotically (as predicted by the complexity analysis), and in terms of its constant factors. It is worth noting that for both ordinary and supersingular inputs, the Magma implementation is substantially slower when working over $\mathbb{F}_{p^2}$ rather than $\mathbb{F}_p$. This is to be expected, given the higher cost of finite field operations in $\mathbb{F}_{p^2}$. By contrast, Algorithm 2 always works in $\mathbb{F}_{p^2}$, and one might suppose that its performance should be essentially independent of whether the input curves is defined over $\mathbb{F}_p$ or $\mathbb{F}_{p^2}$. As indicated by the timings in Table 1, this is not quite the case. There are two reasons for this.

First, for a random elliptic curve $E/\mathbb{F}_{p^2}$, the probability that the vertex $j(E)$ has degree 3 in $G_2(\mathbb{F}_{p^2})$ is, asymptotically, only 1/6. This means that in approximately 5/6 of the cases (whenever $\phi_{\ell,E}(X)$ does not split completely in $\mathbb{F}_{p^2}[X]$), Algorithm 2 terminates in step 3. But if we restrict to $E/\mathbb{F}_p$, this happens in just 1/3 of the cases (namely, whenever $\phi_{\ell,E}(X)$ is irreducible in $\mathbb{F}_p[X]$). This difference explains why Algorithm 2 is actually somewhat faster, on average, when given a random curve over $\mathbb{F}_{p^2}$ rather than $\mathbb{F}_p$.

Second, our implementation relies on a practical optimization that can be applied whenever the input curve is defined over $\mathbb{F}_p$, and this optimization yields nearly a 3-fold speedup on supersingular inputs. Rather than working entirely in the graph $G_2(\mathbb{F}_{p^2})$, we begin by searching for a path in $G_2(\mathbb{F}_p)$ from $j(E)$ to a vertex of degree 1, walking three paths in parallel as usual. Such a vertex j_i will will be found within $O(1)$ steps, on average. The vertex j_i will necessarily have degree 3 in $G_2(\mathbb{F}_{p^2})$, and if E is ordinary, then the two edges that lead from j_i to vertices that are not defined over $\mathbb{F}_p$ must be *descending* edges. It then suffices to extend just one path containing one of these edges, rather than walking three paths in parallel.

6. Acknowledgments

The author is grateful to David Kohel for his feedback on an early draft of this paper, and for showing how to tighten the bound in Proposition 1.

References

[1] Leonard M. Adleman, Kenneth Manders, and Gary L. Miller, *On taking roots in finite fields*, 18th Annual Symposium on Foundations of Computer Science, IEEE, 1977, pp. 175–178.

[2] W. Bosma, J.J. Cannon, C. Fieker, and A. Steel (eds.), *Handbook of Magma functions*, 2.16 ed., 2010, `http://magma.maths.usyd.edu.au/magma/handbook/`.

[3] Reinier Bröker, *Constructing supersingular elliptic curves*, Journal of Combinatorics and Number Theory 1 **1** (2009), no. 3, 269–273.

[4] Noam 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, vol. 7, AMS, 1998, pp. 21–76.

[5] Mireille Fouquet and François Morain, *Isogeny volcanoes and the SEA algorithm*, Algorithmic Number Theory Symposium–ANTS V (C. Fieker and D. R. Kohel, eds.), Lecture Notes in Computer Science, vol. 2369, Springer, 2002, pp. 276–291.

[6] Torbjörn Granlund et al., *GNU multiple precision arithmetic library*, September 2010, version 5.0.1, available at `http://gmplib.org/`.

[7] Dale Husemöller, *Elliptic curves*, Springer-Verlag, 1987.

[8] David Kohel, *Endomorphism rings of elliptic curves over finite fields*, PhD thesis, University of California at Berkeley, 1996.

[9] Serge Lang, *Elliptic functions*, second ed., Springer-Verlag, 1987.

[10] Alfred Menezes, Tatsuaki Okamoto, and Scott A. Vanstone, *Reducing elliptic curve logarithms in a finite field*, IEEE Transactions on Information Theory **39** (1993), 1639–1646.

[11] Gary L. Miller, *Riemann's hypothesis and tests for primality*, Journal of Computer and System Sciences **13** (1976), 300–317.

[12] Arnold K. Pizer, *Ramanujan graphs and Hecke operators*, Bulletin of the American Mathematical Society **23** (1990), no. 1, 127–137.

[13] Hans-Georg Rück, *A note on elliptic curves over finite fields*, Mathematics of Computation **49** (1987), 301–304.

[14] Arnold Schönhage and Volker Strassen, *Schnelle Multiplikation großer Zahlen*, Computing **7** (1971), 281–292.

[15] René Schoof, *Elliptic curves over finite fields and the computation of square roots mod p*, Mathematics of Computation **44** (1985), 483–294.

[16] ———, *Counting points on elliptic curves over finite fields*, Journal de Théorie des Nombres de Bordeaux **7** (1995), 219–254.

[17] ———, *Families of curves and weight distributions of codes*, Bulletin of the American Mathematical Society **32** (1995), no. 2, 171–183.

[18] Donald Shanks, *Five number-theoretic algorithms*, Proceedings of the 2nd Manitoba Conference on Numerical Mathematics, 1972, pp. 51–70.

[19] Joseph H. Silverman, *The arithmetic of elliptic curves*, Springer, 1986.

[20] William A. Stein et al., *Sage Mathematics Software (Version 4.6.2)*, The Sage Development Team, 2011, `http://www.sagemath.org`.

[21] Andrew V. Sutherland, *Computing Hilbert class polynomials with the Chinese Remainder Theorem*, Mathematics of Computation **80** (2011), 501–538.

[22] ———, *Structure computation and discrete logarithms in finite abelian p-groups*, Mathematics of Computation **80** (2011), 477–500.

[23] Alberto Tonelli, *Bemerkung über die Auflösung quadratischer Congruenzen*, Göttinger Nachrichten (1891), 344–346.

[24] Joachim von zur Gathen and Jürgen Gerhard, *Modern computer algebra*, second ed., Cambridge University Press, 2003.

Massachusetts Institute of Technology, Cambridge, Massachusetts 02139
E-mail address: `drew@math.mit.edu`