

A Subexponential Algorithm for Evaluating Large Degree Isogenies

David Jao and Vladimir Soukharev

Department of Combinatorics and Optimization
University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada
{djao,vsoukhar}@math.uwaterloo.ca

Abstract. An isogeny between elliptic curves is an algebraic morphism which is a group homomorphism. Many applications in cryptography require evaluating large degree isogenies between elliptic curves efficiently. For ordinary curves of the same endomorphism ring, the previous best known algorithm has a worst case running time which is exponential in the length of the input. In this paper we show this problem can be solved in subexponential time under reasonable heuristics. Our approach is based on factoring the ideal corresponding to the kernel of the isogeny, modulo principal ideals, into a product of smaller prime ideals for which the isogenies can be computed directly. Combined with previous work of Bostan et al., our algorithm yields equations for large degree isogenies in quasi-optimal time given only the starting curve and the kernel.

1 Introduction

A well known theorem of Tate [29] states that two elliptic curves defined over the same finite field \mathbb{F}_q are isogenous (i.e. admit an isogeny between them) if and only if they have the same number of points over \mathbb{F}_q . Using fast point counting algorithms such as Schoof's algorithm and others [9, 25], it is very easy to check whether this condition holds, and thus whether or not the curves are isogenous. However, constructing the actual isogeny itself is believed to be a hard problem due to the nonconstructive nature of Tate's theorem. Indeed, given an ordinary curve E/\mathbb{F}_q and an ideal of norm n in the endomorphism ring, the fastest previously known algorithm for constructing the unique (up to isomorphism) isogeny having this ideal as kernel has a running time of $O(n^{3+\epsilon})$, except in a certain very small number of special cases [4, 16, 17]. In this paper, we present a new probabilistic algorithm for evaluating such isogenies, which in the vast majority of cases runs (heuristically) in subexponential time. Specifically, we show that for ordinary curves, one can evaluate isogenies of degree n between curves of nearly equal endomorphism ring over \mathbb{F}_q in time less than $L_q(\frac{1}{2}, \frac{\sqrt{3}}{2}) \log(n)$, provided n has no large prime divisors in common with the endomorphism ring discriminant. Although this running time is not polynomial in the input length, our algorithm is still much faster than the (exponential) previous best known algorithm, and in practice allows for the evaluation of isogenies of cryptographically sized degrees, some examples of which we present here. We emphasize that,

in contrast with the previous results of Bröker et al. [4], our algorithm is not limited to special curves such as pairing friendly curves with small discriminant.

If an explicit equation for the isogeny as a rational function is desired, our approach in combination with the algorithm of Bostan et al. [3] can produce the equation in time $O(n^{1+\varepsilon})$ given E and an ideal of norm n , which is quasi-optimal in the sense that (up to log factors) it is equal to the size of the output. To our knowledge, this method is the only known algorithm for computing rational function expressions of large degree isogenies in quasi-optimal time in the general case, given only the starting curve and the kernel.

Apart from playing a central role in the implementation of the point counting algorithms mentioned above, isogenies have been used in cryptography to transfer the discrete logarithm problem from one elliptic curve to another [9, 16, 17, 20, 23, 30]. In many of these applications, our algorithm cannot be used directly, since in cryptography one is usually given two isogenous curves, rather than one curve together with the isogeny degree. However, earlier results [16, 17, 20] have shown that the problem of computing isogenies between a given pair of curves can be reduced to the problem of computing isogenies of prime degree starting from a given curve. It is therefore likely that the previous best isogeny construction algorithms in the cryptographic setting can be improved or extended in light of the work that we present here.

2 Background

Let E and E' be elliptic curves defined over a finite field \mathbb{F}_q of characteristic p . An isogeny $\phi: E \rightarrow E'$ defined over \mathbb{F}_q is a non-constant rational map defined over \mathbb{F}_q which is also a group homomorphism from $E(\mathbb{F}_q)$ to $E'(\mathbb{F}_q)$. This definition differs slightly from the standard definition in that it excludes constant maps [27, §III.4]. The degree of an isogeny is its degree as a rational map, and an isogeny of degree ℓ is called an ℓ -isogeny. Every isogeny of degree greater than 1 can be factored into a composition of isogenies of prime degree defined over $\overline{\mathbb{F}}_q$ [11].

For any elliptic curve $E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ defined over \mathbb{F}_q , the Frobenius endomorphism is the isogeny $\pi_q: E \rightarrow E$ of degree q given by the equation $\pi_q(x, y) = (x^q, y^q)$. The characteristic polynomial of π_q is $X^2 - tX + q$ where $t = q + 1 - \#E(\mathbb{F}_q)$ is the trace of E .

An endomorphism of E is an isogeny $E \rightarrow E$ defined over the algebraic closure $\overline{\mathbb{F}}_q$ of \mathbb{F}_q . The set of endomorphisms of E together with the zero map forms a ring under the operations of pointwise addition and composition; this ring is called the endomorphism ring of E and denoted $\text{End}(E)$. The ring $\text{End}(E)$ is isomorphic either to an order in a quaternion algebra or to an order in an imaginary quadratic field [27, V.3.1]; in the first case we say E is supersingular and in the second case we say E is ordinary.

Two elliptic curves E and E' defined over \mathbb{F}_q are said to be isogenous over \mathbb{F}_q if there exists an isogeny $\phi: E \rightarrow E'$ defined over \mathbb{F}_q . A theorem of Tate states that two curves E and E' are isogenous over \mathbb{F}_q if and only if $\#E(\mathbb{F}_q) = \#E'(\mathbb{F}_q)$ [29, §3]. Since every isogeny has a dual isogeny [27, III.6.1], the property of being

isogenous over \mathbb{F}_q is an equivalence relation on the finite set of $\overline{\mathbb{F}}_q$ -isomorphism classes of elliptic curves defined over \mathbb{F}_q . Moreover, isomorphisms between elliptic curves can be classified completely and computed efficiently in all cases [16]. Accordingly, we define an isogeny class to be an equivalence class of elliptic curves, taken up to $\overline{\mathbb{F}}_q$ -isomorphism, under this equivalence relation.

Curves in the same isogeny class are either all supersingular or all ordinary. The vast majority of curves are ordinary, and indeed the number of isomorphism classes of supersingular curves is finite for each characteristic. Also, ordinary curves form the majority of the curves of interest in applications such as cryptography. Hence, we assume for the remainder of this paper that we are in the **ordinary case**.

Let K denote the imaginary quadratic field containing $\text{End}(E)$, with maximal order \mathcal{O}_K . For any order $\mathcal{O} \subseteq \mathcal{O}_K$, the conductor of \mathcal{O} is defined to be the integer $[\mathcal{O}_K : \mathcal{O}]$. The field K is called the CM field of E . We write c_E for the conductor of $\text{End}(E)$ and c_π for the conductor of $\mathbb{Z}[\pi_q]$. It follows from [12, §7] that $\text{End}(E) = \mathbb{Z} + c_E \mathcal{O}_K$ and $\Delta = c_E^2 \Delta_K$, where Δ (respectively, Δ_K) is the discriminant of the imaginary quadratic order $\text{End}(E)$ (respectively, \mathcal{O}_K). Furthermore, the characteristic polynomial has discriminant $\Delta_\pi = t^2 - 4q = \text{disc}(\mathbb{Z}[\pi_q]) = c_\pi^2 \Delta_K$, with $c_\pi = c_E \cdot [\text{End}(E) : \mathbb{Z}[\pi_q]]$.

Following [14] and [16], we say that an isogeny $\phi: E \rightarrow E'$ of prime degree ℓ defined over \mathbb{F}_q is “down” if $[\text{End}(E) : \text{End}(E')] = \ell$, “up” if $[\text{End}(E') : \text{End}(E)] = \ell$, and “horizontal” if $\text{End}(E) = \text{End}(E')$. Two curves in an isogeny class are said to “have the same level” if their endomorphism rings are equal. Within each isogeny class, the property of having the same level is an equivalence relation. A horizontal isogeny always goes between two curves of the same level; likewise, an up isogeny enlarges the endomorphism ring and a down isogeny reduces it. Since there are fewer elliptic curves at higher levels than at lower levels, the collection of elliptic curves in an isogeny class visually resembles a “pyramid” or a “volcano” [14], with up isogenies ascending the structure and down isogenies descending. If we restrict to the graph of ℓ -isogenies for a single ℓ , then in general the ℓ -isogeny graph is disconnected, having one ℓ -volcano for each intermediate order $\mathbb{Z}[\pi_q] \subset \mathcal{O} \subset \mathcal{O}_K$ such that \mathcal{O} is maximal at ℓ (meaning $\ell \nmid [\mathcal{O}_K : \mathcal{O}]$). The “top level” of the class consists of curves E with $\text{End}(E) = \mathcal{O}_K$, and the “bottom level” consists of curves with $\text{End}(E) = \mathbb{Z}[\pi_q]$.

We say that ℓ is an *Elkies prime* [2, p. 119] if $\ell \nmid c_E$ and $(\frac{\Delta}{\ell}) \neq -1$, or equivalently if and only if E admits a horizontal isogeny of degree ℓ . The number of ℓ -isogenies of each type can easily be determined explicitly [14, 16, 21]. In particular, for all but the finitely many primes ℓ dividing $[\mathcal{O}_K : \mathbb{Z}[\pi_q]]$, we have that every rational ℓ -isogeny admitted by E is horizontal.

3 The Bröker-Charles-Lauter algorithm

Our algorithm is an extension of the algorithm developed by Bröker, Charles, and Lauter [4] to evaluate large degree isogenies over ordinary elliptic curves with

endomorphism rings of small class number, such as pairing-friendly curves [15]. In this section we provide a summary of their results.

The following notation corresponds to that of [4]. Let E/\mathbb{F}_q be an ordinary elliptic curve with endomorphism ring $\text{End}(E)$ isomorphic to an imaginary quadratic order \mathcal{O}_Δ of discriminant $\Delta < 0$. Identify $\text{End}(E)$ with \mathcal{O}_Δ via the unique isomorphism ι such that $\iota^*(x)\omega = x\omega$ for all invariant differentials ω and all $x \in \mathcal{O}_\Delta$. Then every horizontal separable isogeny on E of prime degree ℓ corresponds (up to isomorphism) to a unique prime ideal $\mathfrak{L} \subset \mathcal{O}_\Delta$ of norm ℓ for some Elkies prime ℓ . We denote the kernel of this isogeny by $E[\mathfrak{L}]$. Any two distinct isomorphic horizontal isogenies (i.e., pairs of isogenies where one is equal to the composition of the other with an isomorphism) induce different maps on the space of differentials of E , and a separable isogeny is uniquely determined by the combination of its kernel and the induced map on the space of differentials. A *normalized* isogeny is an isogeny $\phi: E \rightarrow E'$ for which $\phi^*(\omega_{E'}) = \omega_E$ where ω_E denotes the invariant differential of E . Algorithm 1 (identical to Algorithm 4.1 in [4]) evaluates, up to automorphisms of E , the unique normalized horizontal isogeny of degree ℓ corresponding to a given kernel ideal $\mathfrak{L} \subset \mathcal{O}_\Delta$.

The following theorem, taken verbatim from [4], shows that the running time of Algorithm 1 is polynomial in the quantities $\log(\ell)$, $\log(q)$, n , and $|\Delta|$.

Theorem 3.1. *Let E/\mathbb{F}_q be an ordinary elliptic curve with Frobenius π_q , given by a Weierstrass equation, and let $P \in E(\mathbb{F}_{q^n})$ be a point on E . Let $\Delta = \text{disc}(\text{End}(E))$ be given. Assume that $[\text{End}(E) : \mathbb{Z}[\pi_q]]$ and $\#E(\mathbb{F}_{q^n})$ are coprime, and let $\mathfrak{L} = (\ell, c + d\pi_q)$ be an $\text{End}(E)$ -ideal of prime norm $\ell \neq \text{char}(\mathbb{F}_q)$ not dividing the index $[\text{End}(E) : \mathbb{Z}[\pi_q]]$. Algorithm 1 computes the unique elliptic curve E' such that there exists a normalized isogeny $\phi: E \rightarrow E'$ with kernel $E[\mathfrak{L}]$. Furthermore, it computes the x -coordinate of $\phi(P)$ if $\text{End}(E)$ does not equal $\mathbb{Z}[i]$ or $\mathbb{Z}[\zeta_3]$ and the square, respectively cube, of the x -coordinate of $\phi(P)$ otherwise. The running time of the algorithm is polynomial in $\log(\ell)$, $\log(q)$, n and $|\Delta|$.*

4 A subexponential algorithm for evaluating horizontal isogenies

As was shown in Sections 2 and 3, any horizontal isogeny can be expressed as a composition of prime degree isogenies, one for each prime factor of the kernel, and any prime degree isogeny is a composition of a normalized isogeny and an isomorphism. Therefore, to evaluate a horizontal isogeny given its kernel, it suffices to treat the case of horizontal normalized prime degree isogenies.

Our objective is to evaluate the unique horizontal normalized isogeny on a given elliptic curve E/\mathbb{F}_q whose kernel ideal is given as $\mathfrak{L} = (\ell, c + d\pi_q)$, at a given point $P \in E(\mathbb{F}_{q^n})$, where ℓ is an Elkies prime. As in [4], we must also impose the additional restriction that $\ell \nmid [\text{End}(E) : \mathbb{Z}[\pi_q]]$; for Elkies primes, an equivalent restriction is that $\ell \nmid [\mathcal{O}_K : \mathbb{Z}[\pi_q]]$, but we retain the original formulation for consistency with [4].

Algorithm 1 The Bröker-Charles-Lauter algorithm

- Input:** A discriminant Δ , an elliptic curve E/\mathbb{F}_q with $\text{End}(E) = \mathcal{O}_\Delta$ and a point $P \in E(\mathbb{F}_{q^n})$ such that $[\text{End}(E) : \mathbb{Z}[\pi_q]]$ and $\#E(\mathbb{F}_{q^n})$ are coprime, and an $\text{End}(E)$ -ideal $\mathfrak{L} = (\ell, c + d\pi_q)$ of prime norm $\ell \neq \text{char}(\mathbb{F}_q)$ not dividing the index $[\text{End}(E) : \mathbb{Z}[\pi_q]]$.
- Output:** The unique elliptic curve E' admitting a normalized isogeny $\phi : E \rightarrow E'$ with kernel $E[\mathfrak{L}]$, and the x -coordinate of $\phi(P)$ for $\Delta \neq -3, -4$ and the square (resp. cube) of the x -coordinate otherwise.
- 1: Compute the direct sum decomposition $\text{Pic}(\mathcal{O}_\Delta) = \bigotimes ([I_i])$ of $\text{Pic}(\mathcal{O}_\Delta)$ into cyclic groups generated by the degree 1 prime ideals I_i of smallest norm that are coprime to the product $p \cdot \#E(\mathbb{F}_{q^n}) \cdot [\text{End}(E) : \mathbb{Z}[\pi_q]]$.
 - 2: Using brute force¹, find e_1, e_2, \dots, e_k such that $[\mathfrak{L}] = [I_1^{e_1}] \cdot [I_2^{e_2}] \cdots [I_k^{e_k}]$.
 - 3: Find α (using Cornacchia’s algorithm) and express $\mathfrak{L} = I_1^{e_1} \cdot I_2^{e_2} \cdots I_k^{e_k} \cdot (\alpha)$.
 - 4: Compute a sequence of isogenies (ϕ_1, \dots, ϕ_s) such that the composition $\phi_c : E \rightarrow E_c$ has kernel $E[I_1^{e_1} \cdot I_2^{e_2} \cdots I_k^{e_k}]$ using the method of [4, § 3].
 - 5: Evaluate $\phi_c(P) \in E_c(\mathbb{F}_{q^n})$.
 - 6: Write $\alpha = (u + v\pi_q)/(zm)$. Compute the isomorphism $\eta : E_c \xrightarrow{\sim} E'$ with $\eta^*(\omega_{E'}) = (u/zm)\omega_{E_c}$. Compute $Q = \eta(\phi_c(P))$.
 - 7: Compute $(zm)^{-1} \bmod \#E(\mathbb{F}_{q^n})$, and compute $R = ((zm)^{-1}(u + v\pi_q))(Q)$.
 - 8: Put $r = x(R)^{|\mathcal{O}_\Delta|^*/2}$ and return (E', r) .
-

In practice, one is typically given ℓ instead of \mathfrak{L} , but since it is easy to calculate the list of (at most two) possible primes \mathfrak{L} lying over ℓ (cf. [6]), these two interpretations are for all practical purposes equivalent, and we switch freely between them when convenient. When ℓ is small, one can use modular polynomial based techniques [4, §3.1], which have running time $O(\ell^3 \log(\ell)^{4+\epsilon})$ [13]. However, for isogeny degrees of cryptographic size (e.g. 2^{160}), this approach is impractical. The Bröker-Charles-Lauter algorithm sidesteps this problem, by using an alternative factorization of \mathfrak{L} . However, the running time of Bröker-Charles-Lauter is polynomial in $|\Delta|$, and therefore even this method only works for small values of $|\Delta|$. In this section we present a modified version of the Bröker-Charles-Lauter algorithm which is suitable for large values of $|\Delta|$.

We begin by giving an overview of our approach. In order to handle large values of $|\Delta|$, there are two main problems to overcome. One problem is that we need a fast way to produce a factorization

$$\mathfrak{L} = I_1^{e_1} I_2^{e_2} \cdots I_k^{e_k} \cdot (\alpha) \tag{1}$$

as in lines 2 and 3 of Algorithm 1. The other problem is that the exponents e_i in Equation (1) need to be kept small, since the running times of lines 3 and 4 of Algorithm 1 are proportional to $\sum_i |e_i| \text{Norm}(I_i)^2$. The first problem, that of finding a factorization of \mathfrak{L} , can be solved in subexponential time using the index calculus algorithm of Hafner and McCurley [18] (see also [6, Chap. 11]).

¹ Bröker, Charles, and Lauter mention that this computation can be done in “various ways” [4, p. 107], but the only explicit method given in [4] is brute force. The use of brute force limits the algorithm to elliptic curves for which $|\Delta|$ is small, such as pairing-friendly curves.

Algorithm 2 Computing a factor base

Input: A discriminant Δ , a bound N .**Output:** The set \mathcal{I} consisting of split prime ideals of norm less than N , together with the corresponding set \mathcal{F} of quadratic forms.

- 1: Set $\mathcal{F} \leftarrow \emptyset$.
 - 2: Set $\mathcal{I} \leftarrow \emptyset$.
 - 3: Find all primes $p < N$ such that $(\frac{\Delta}{p}) = 1$. Call this set P . Let $k = |P|$.
 - 4: For each prime $p_i \in P$, find an ideal \mathfrak{p}_i of norm p_i (using Cornacchia's algorithm).
 - 5: For each i , find a quadratic form $f_i = [(p_i, b_i, c_i)]$ corresponding to \mathfrak{p}_i in $\text{Cl}(\mathcal{O}_\Delta)$, using the technique of [26, §3].
 - 6: Output $\mathcal{I} = \{\mathfrak{p}_1, \mathfrak{p}_2, \dots, \mathfrak{p}_k\}$ and $\mathcal{F} = \{f_1, f_2, \dots, f_k\}$.
-

To resolve the second problem, we turn to an idea which was first introduced by Galbraith et. al [17], and recently further refined by Bisson and Sutherland [1]. The idea is that, in the process of sieving for smooth norms, one can arbitrarily restrict the input exponent vectors to sparse vectors (e_1, e_2, \dots, e_k) such that $\sum_i |e_i| N(I_i)^2$ is kept small. This restriction is implemented in line 6 of Algorithm 3. As in [1], one then assumes heuristically that the imposition of this restriction does not affect the eventual probability of obtaining a smooth norm in the Hafner and McCurley algorithm. Note that, unlike the input exponents, the exponents appearing in the factorizations of the ensuing smooth norms (that is, the values of y_i in Algorithm 3) are always small, since the norm in question is derived from a reduced quadratic form.

We now describe the individual components of our algorithm in detail.

4.1 Finding a factor base

Let $\text{Cl}(\mathcal{O}_\Delta)$ denote the ideal class group of \mathcal{O}_Δ . Algorithm 2 produces a factor base consisting of split primes in \mathcal{O}_Δ of norm less than some bound N . The optimal value of N will be determined in Section 4.4.

4.2 “Factoring” large prime degree ideals

Algorithm 3, based on the algorithm of Hafner and McCurley, takes as input a discriminant Δ , a curve E , a prime ideal \mathfrak{L} of prime norm ℓ in \mathcal{O}_Δ , a smoothness bound N , and an extension degree n . It outputs a factorization

$$\mathfrak{L} = I_1^{e_1} I_2^{e_2} \cdots I_k^{e_k} \cdot (\alpha)$$

as in Equation 1, where the I_i 's are as in Algorithm 1, the exponents e_i are positive, sparse, and small (i.e., polynomial in N), and the ideal (α) is a principal fractional ideal generated by α .

4.3 Algorithm for evaluating prime degree isogenies

The overall algorithm for evaluating prime degree isogenies is given in Algorithm 4. This algorithm is identical to Algorithm 1, except that the factorization of \mathfrak{L} is performed using Algorithm 3. To maintain consistency with [4], we

Algorithm 3 “Factoring” a prime ideal

Input: A discriminant Δ , an elliptic curve E/\mathbb{F}_q with $\text{End}(E) = \mathcal{O}_\Delta$, a smoothness bound N , a prime ideal \mathfrak{L} of norm ℓ in \mathcal{O}_Δ , an extension degree n .

Output: Relation of the form $\mathfrak{L} = (\alpha) \cdot \prod_{i=1}^k I_i^{e_i}$, where (α) is a fractional ideal, I_i are as in Algorithm 1, and $e_i > 0$ are small and sparse.

- 1: Run Algorithm 2 on input Δ and N to obtain $\mathcal{I} = \{\mathfrak{p}_1, \mathfrak{p}_2, \dots, \mathfrak{p}_k\}$ and $\mathcal{F} = \{f_1, f_2, \dots, f_k\}$. Discard any primes dividing $p \cdot \#E(\mathbb{F}_{q^n}) \cdot [\text{End}(E) : \mathbb{Z}[\pi_q]]$.
 - 2: Set $p_i \leftarrow \text{Norm}(\mathfrak{p}_i)$. (These values are also calculated in Algorithm 2.)
 - 3: Obtain the reduced quadratic form $[\mathfrak{L}]$ corresponding to the ideal class of \mathfrak{L} .
 - 4: **repeat**
 - 5: **for** $i = 1, \dots, k$ **do**
 - 6: Pick exponents x_i in the range $[0, (N/p_i)^2]$ such that at most k_0 are nonzero, where k_0 is a global absolute constant (in practice, $k_0 = 3$ suffices).
 - 7: **end for**
 - 8: Compute the reduced quadratic form $\mathfrak{a} = (a, b, c)$ for which the ideal class $[\mathfrak{a}]$ is equivalent to $[\mathfrak{L}] \cdot \prod_{i=1}^k f_i^{x_i}$.
 - 9: **until** The integer a factors completely into the primes p_i , and the relation derived from $[\mathfrak{a}] = [\mathfrak{L}] \cdot \prod_{i=1}^k f_i^{x_i}$ contains fewer than $\sqrt{\log(|\Delta|/3)}/z$ nonzero exponents.
 - 10: Write $a = \prod_{i=1}^k p_i^{u_i}$.
 - 11: **for** $i=1, \dots, k$ **do**
 - 12: Using the technique of Seysen ([26, Theorem 3.1]), determine the signs of the exponents $y_i = \pm u_i$ for which $\mathfrak{a} = \prod_{i=1}^k f_i^{y_i}$.
 - 13: Let $e_i = y_i - x_i$. (These exponents satisfy $[\mathfrak{L}] = \prod_{i=1}^k f_i^{e_i}$.)
 - 14: **if** $e_i \geq 0$ **then**
 - 15: Set $I_i \leftarrow \bar{\mathfrak{p}}_i$
 - 16: **else**
 - 17: Set $I_i \leftarrow \mathfrak{p}_i$
 - 18: **end if**
 - 19: **end for**
 - 20: Compute the principal ideal $I = \mathfrak{L} \cdot \prod_{i=1}^k I_i^{|e_i|}$.
 - 21: Using Cornacchia’s algorithm, find a generator $\beta \in \mathcal{O}_\Delta$ of I .
 - 22: Set $m \leftarrow \prod_{i=1}^k p_i^{|e_i|}$ and $\alpha \leftarrow \frac{\beta}{m}$.
 - 23: Output $\mathfrak{L} = (\alpha) \cdot \bar{I}_1^{|e_1|} \cdot \bar{I}_2^{|e_2|} \dots \bar{I}_k^{|e_k|}$.
-

have included the quantities Δ and $\text{End}(E)$ as part of the input to the algorithm. However, we remark that these quantities can be computed from E/\mathbb{F}_q in $L_q(\frac{1}{2}, \frac{\sqrt{3}}{2})$ operations using the algorithm of Bisson and Sutherland [1], even if they are not provided as input.

4.4 Running time analysis

In this section, we determine the theoretical running time of Algorithm 4, as well as the optimal value of the smoothness bound N to use in line 1 of the algorithm. As is typical for subexponential time factorization algorithms involving a factor base, these two quantities depend on each other, and hence both are calculated simultaneously.

Algorithm 4 Evaluating prime degree isogenies

Input: A discriminant Δ , an elliptic curve E/\mathbb{F}_q with $\text{End}(E) = \mathcal{O}_\Delta$ and a point $P \in E(\mathbb{F}_{q^n})$ such that $[\text{End}(E) : \mathbb{Z}[\pi_q]]$ and $\#E(\mathbb{F}_{q^n})$ are coprime, and an $\text{End}(E)$ -ideal $\mathfrak{L} = (\ell, c + d\pi_q)$ of prime norm $\ell \neq \text{char}(\mathbb{F}_q)$ not dividing the index $[\text{End}(E) : \mathbb{Z}[\pi_q]]$.

Output: The unique elliptic curve E' admitting a normalized isogeny $\phi: E \rightarrow E'$ with kernel $E[\mathfrak{L}]$, and the x -coordinate of $\phi(P)$ for $\Delta \neq -3, -4$ and the square (resp. cube) of the x -coordinate otherwise.

- 1: Choose a smoothness bound N (see Section 4.4).
- 2: Using Algorithm 3 on input $(\Delta, E, N, \mathfrak{L}, n)$, obtain a factorization of the form $\mathfrak{L} = I_1^{e_1} \cdot I_2^{e_2} \cdots I_k^{e_k} \cdot (\alpha)$.
- 3: Compute a sequence of isogenies (ϕ_1, \dots, ϕ_s) such that the composition $\phi_c: E \rightarrow E_c$ has kernel $E[I_1^{e_1} \cdot I_2^{e_2} \cdots I_k^{e_k}]$ using the method of [4, § 3].
- 4: Evaluate $\phi_c(P) \in E_c(\mathbb{F}_{q^n})$.
- 5: Write $\alpha = (u + v\pi_q)/(zm)$. Compute the isomorphism $\eta: E_c \xrightarrow{\sim} E'$ with $\eta^*(\omega_{E'}) = (u/zm)\omega_{E_c}$. Compute $Q = \eta(\phi_c(P))$.
- 6: Compute $(zm)^{-1} \bmod \#E(\mathbb{F}_{q^n})$, and compute $R = ((zm)^{-1}(u + v\pi_q))(Q)$.
- 7: Put $r = x(R)^{|\mathcal{O}_\Delta|^*/2}$ and return (E', r) .

As in [9], we define² $L_n(\alpha, c)$ by

$$L_n(\alpha, c) = O(\exp((c + o(1))(\log(n))^\alpha (\log(\log(n)))^{1-\alpha})).$$

The quantity $L_n(\alpha, c)$ interpolates between polynomial and exponential size as α ranges from 0 to 1. We set $N = L_{|\Delta|}(\frac{1}{2}, z)$ for an unspecified value of z , and in the following paragraphs we determine the optimal value of z which minimizes the running time of Algorithm 4. (The fact that $\alpha = \frac{1}{2}$ is optimal is clear from the below analysis, as well as from prior experience with integer factorization algorithms.) For convenience, we will abbreviate $L_{|\Delta|}(\alpha, c)$ to $L(\alpha, c)$ throughout.

Line 2 of Algorithm 4 involves running Algorithm 3, which in turn calls Algorithm 2. As it turns out, Algorithm 2 is almost the same as Algorithm 11.1 from [6], which requires $L(\frac{1}{2}, z)$ time, as shown in [6]. The only difference is that we add an additional step where we obtain the quadratic form corresponding to each prime ideal in the factor base. This extra step requires $O(\log(\text{Norm}(I))^{1+\varepsilon})$ time for a prime ideal I , using Cornacchia's Algorithm [19]. Thus, the overall running time for Algorithm 2 is bounded above by

$$L(\frac{1}{2}, z) \cdot \log(L(\frac{1}{2}, z))^{1+\varepsilon} = L(\frac{1}{2}, z).$$

Line 2 of Algorithm 3 takes $\log(\ell)$ time using standard algorithms [12]. The loop in lines 4–9 of Algorithm 3 is very similar to the FINDRELATION algorithm in [1], except that we only use one discriminant, and we omit the requirement that $\#R/D_1 > \#R/D_2$ (which in any case is meaningless when there is only one discriminant). Needless to say, this change can only speed up the algorithm. Taking $\mu = \sqrt{2}z$ in [1, Prop. 6], we find that the (heuristic) expected running time of the loop in lines 4–9 of Algorithm 3 is $L(\frac{1}{2}, \frac{1}{4z})$.

² The definition of $L_n(\alpha, c)$ in [6] differs from that of [9] in the $o(1)$ term. We account for this discrepancy in our text.

The next step in Algorithm 3 having nontrivial running time is the computation of the ideal product in line 20. To exponentiate an element of an arbitrary semigroup to a power e requires $O(\log e)$ semigroup multiplication operations [10, §1.2]. To multiply two ideals I and J in an imaginary quadratic order (via composition of quadratic forms) requires $O(\max(\log(\text{Norm}(I)), \log(\text{Norm}(J)))^{1+\varepsilon})$ bit operations using fast multiplication [24, §6]. Each of the expressions $|I_i|^{|e_i|}$ therefore requires $O(\log |e_i|)$ ideal multiplication operations to compute, with each individual multiplication requiring

$$O((|e_i| \log(\text{Norm}(I_i)))^{1+\varepsilon}) = O\left(\left(\left(\frac{N}{p_i}\right)^2 \log(p_i)\right)^{1+\varepsilon}\right) = O(N^{2+\varepsilon})$$

bit operations, for a total running time of $(\log e_i)O(N^{2+\varepsilon}) = L(\frac{1}{2}, 2z)$ for each i . This calculation must be performed once for each nonzero exponent e_i . By line 9, the number of nonzero exponents appearing in the relation is at most $\sqrt{\log(|\Delta|/3)}/z$, so the amount of time required to compute all of the $|I_i|^{|e_i|}$ for all i is $(\sqrt{\log(|\Delta|/3)}/z)L(\frac{1}{2}, 2z) = L(\frac{1}{2}, 2z)$. Afterward, the values $|I_i|^{|e_i|}$ must all be multiplied together, a calculation which entails at most $\sqrt{\log(|\Delta|/3)}/z$ ideal multiplications where the log-norms of the input multiplicands are bounded above by

$$\log \text{Norm}(I_i^{|e_i|}) = |e_i| \log \text{Norm}(I_i) \leq \left(\frac{N}{p_i}\right)^2 \log p_i \leq N^2 = L(\frac{1}{2}, 2z),$$

and thus each of the (at most) $\sqrt{\log(|\Delta|/3)}/z$ multiplications in the ensuing product can be completed in time at most $(\sqrt{\log(|\Delta|/3)}/z)L(\frac{1}{2}, 2z) = L(\frac{1}{2}, 2z)$. Finally, we must multiply this end result by \mathfrak{L} , an operation which requires $O(\max(\log \ell, L(\frac{1}{2}, 2z))^{1+\varepsilon})$ time. All together, the running time of step 20 is $L(\frac{1}{2}, 2z) + O(\max(\log \ell, L(\frac{1}{2}, 2z))^{1+\varepsilon}) = \max((\log \ell)^{1+\varepsilon}, L(\frac{1}{2}, 2z))$, and the norm of the resulting ideal I is bounded above by $\ell \cdot \exp(L(\frac{1}{2}, 2z))$.

Obtaining the generator β of I in line 21 of Algorithm 3 using Cornacchia's algorithm requires

$$O(\log(\text{Norm}(I))^{1+\varepsilon}) = (\log \ell + L(\frac{1}{2}, 2z))^{1+\varepsilon}$$

time. We remark that finding β given I is substantially easier than the usual Cornacchia's algorithm, which entails finding β given only $\text{Norm}(I)$. The usual algorithm requires finding *all* the square roots of Δ modulo $\text{Norm}(I)$, which is very slow when $\text{Norm}(I)$ has a large number of prime divisors. This time-consuming step is unnecessary when the ideal I itself is given, since the embedding of the ideal I in $\text{End}(E)$ already provides (up to sign) the correct square root of Δ mod I . A detailed description of this portion of Cornacchia's algorithm in the context of the full algorithm, together with running time figures specific to each sub-step, is given by Hardy et al. [19]; for our purposes, the running time of a single iteration of Step 6 in [19, §4] is the relevant figure. This concludes our analysis of Algorithm 3.

Returning to Algorithm 4, we find that (as in [4]) the computation of the individual isogenies ϕ_i in line 3 of Algorithm 4 is limited by the time required to compute the modular polynomials $\Phi_n(x, y)$. Using the Chinese remainder theorem-based method of Bröker et al. [5], these polynomials can be computed mod q in time $O(n^3 \log^{3+\varepsilon}(n))$, and the resulting polynomials require $O(n^2(\log^2 n + \log q))$ space. For each ideal I_i , the corresponding modular polynomial of level p_i only needs to be computed once, but the polynomial once computed must be evaluated, differentiated, and otherwise manipulated e_i times, at a cost of $O(p_i^{2+\varepsilon})$ field operations in \mathbb{F}_q per manipulation, or $O(p_i^{2+\varepsilon})(\log q)^{1+\varepsilon}$ bit operations using fast multiplication. The total running time of line 3 is therefore

$$\begin{aligned} O(p_i^{3+\varepsilon}) + \sum_i |e_i| p_i^{2+\varepsilon} (\log q)^{1+\varepsilon} &\leq O(N^{3+\varepsilon}) + \sum_i \left(\left(\frac{N}{p_i} \right)^2 \right) p_i^{2+\varepsilon} (\log q)^{1+\varepsilon} \\ &\leq O(N^{3+\varepsilon}) + \frac{\sqrt{\log(|\Delta|/3)}}{z} N^{2+\varepsilon} (\log q)^{1+\varepsilon} = L\left(\frac{1}{2}, 3z\right) + L\left(\frac{1}{2}, 2z\right) (\log q)^{1+\varepsilon}. \end{aligned}$$

Similarly, the evaluation of ϕ_c in line 4 requires

$$\sum_i |e_i| p_i^{2+\varepsilon} = L\left(\frac{1}{2}, 2z\right)$$

field operations in \mathbb{F}_{q^n} , which corresponds to $L\left(\frac{1}{2}, 2z\right) (\log q^n)^{1+\varepsilon}$ bit operations using fast multiplication.

Combining all the above quantities, we obtain a total running time of

$$\begin{aligned} &L\left(\frac{1}{2}, z\right) && \text{(algorithm 2)} \\ &+ L\left(\frac{1}{2}, \frac{1}{4z}\right) && \text{(lines 4–9, algorithm 3)} \\ &+ \max((\log \ell)^{1+\varepsilon}, L\left(\frac{1}{2}, 2z\right)) && \text{(line 20, algorithm 3)} \\ &+ (\log \ell + L\left(\frac{1}{2}, 2z\right))^{1+\varepsilon} && \text{(line 21, algorithm 3)} \\ &+ L\left(\frac{1}{2}, 3z\right) + L\left(\frac{1}{2}, 2z\right) (\log q)^{1+\varepsilon} && \text{(line 3, algorithm 4)} \\ &+ L\left(\frac{1}{2}, 2z\right) (\log q^n)^{1+\varepsilon} && \text{(line 4, algorithm 4)} \end{aligned}$$

$$= L\left(\frac{1}{2}, \frac{1}{4z}\right) + (\log \ell + L\left(\frac{1}{2}, 2z\right))^{1+\varepsilon} + L\left(\frac{1}{2}, 3z\right) + L\left(\frac{1}{2}, 2z\right) (\log q^n)^{1+\varepsilon}.$$

When $|\Delta|$ is large, we may impose the reasonable assumption that $\log(\ell) \ll L\left(\frac{1}{2}, z\right)$ and $\log(q^n) \ll L\left(\frac{1}{2}, z\right)$. In this case, the running time of Algorithm 4 is dominated by the expression $L\left(\frac{1}{2}, \frac{1}{4z}\right) + L\left(\frac{1}{2}, 3z\right)$, which attains a minimum at $z = \frac{1}{2\sqrt{3}}$. Taking this value of z , we find that the running time of Algorithm 4 is equal to $L_{|\Delta|}\left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right)$. Since the maximum value of $|\Delta| \leq |\Delta_\pi| = 4q - t^2$ is $4q$, we can alternatively express this running time as simply $L_q\left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right)$.

In the general case, $\log(\ell)$ and $\log(q^n)$ might be non-negligible compared to $L\left(\frac{1}{2}, z\right)$. This can happen in one of two ways: either $|\Delta|$ is small, or (less likely)

ℓ is very large and/or n is large. When this happens, we can still bound the running time of Algorithm 4 by taking $z = \frac{1}{2\sqrt{3}}$ in the foregoing calculation, although such a choice may fail to be optimal. We then find that the running time of Algorithm 4 is bounded above by

$$(\log(\ell) + L(\frac{1}{2}, \frac{1}{\sqrt{3}}))^{1+\varepsilon} + L(\frac{1}{2}, \frac{\sqrt{3}}{2}) + L(\frac{1}{2}, \frac{1}{\sqrt{3}})(\log q^n)^{1+\varepsilon}.$$

We summarize our results in the following theorem.

Theorem 4.1. *Let E/\mathbb{F}_q be an ordinary elliptic curve with Frobenius π_q , given by a Weierstrass equation, and let $P \in E(\mathbb{F}_{q^n})$ be a point on E . Let $\Delta = \text{disc}(\text{End}(E))$ be given. Assume that $[\text{End}(E) : \mathbb{Z}[\pi_q]]$ and $\#E(\mathbb{F}_{q^n})$ are coprime, and let $\mathfrak{L} = (\ell, c + d\pi_q)$ be an $\text{End}(E)$ -ideal of prime norm $\ell \neq \text{char}(\mathbb{F}_q)$ not dividing the index $[\text{End}(E) : \mathbb{Z}[\pi_q]]$. Under the heuristics of [1, §4], Algorithm 4 computes the unique elliptic curve E' such that there exists a normalized isogeny $\phi: E \rightarrow E'$ with kernel $E[\mathfrak{L}]$. Furthermore, it computes the x -coordinate of $\phi(P)$ if $\text{End}(E)$ does not equal $\mathbb{Z}[i]$ or $\mathbb{Z}[\zeta_3]$ and the square, respectively cube, of the x -coordinate of $\phi(P)$ otherwise. The running time of the algorithm is bounded above by*

$$(\log(\ell) + L(\frac{1}{2}, \frac{1}{\sqrt{3}}))^{1+\varepsilon} + L(\frac{1}{2}, \frac{\sqrt{3}}{2}) + L(\frac{1}{2}, \frac{1}{\sqrt{3}})(\log q^n)^{1+\varepsilon}.$$

The running time of the algorithm is subexponential in $\log |\Delta|$, and polynomial in $\log(\ell)$, $\log(q)$, and n .

5 Examples

5.1 Small example

Let $p = 10^{10} + 19$ and let E/\mathbb{F}_p be the curve $y^2 = x^3 + 15x + 129$. Then $E(\mathbb{F}_p)$ has cardinality $10000036491 = 3 \cdot 3333345497$ and trace $t = -36471$. To avoid any bias in the selection of the prime ℓ , we set ℓ to be the smallest Elkies prime of E larger than $p/2$, namely $\ell = 5000000029$. We will evaluate the x -coordinate of $\phi(P)$, where ϕ is an isogeny of degree ℓ , and P is chosen arbitrarily to be the point $(5940782169, 2162385016) \in E(\mathbb{F}_p)$. We remark that, although this example is designed to be artificially small for illustration purposes, the evaluation of this isogeny would already be infeasible if we were using prior techniques based on modular functions of level ℓ .

The discriminant Δ of E is $\Delta = t^2 - 4p = -38669866235$. Set $w = \frac{1+\sqrt{\Delta}}{2}$ and $\mathcal{O} = \mathcal{O}_\Delta$. The quadratic form $(5000000029, -2326859861, 270713841)$ represents a prime ideal \mathfrak{L} of norm ℓ , and we show how to calculate the isogeny ϕ having kernel corresponding to $E[\mathfrak{L}]$. Using an implementation of Algorithm 3 in MAGMA [22], we find immediately the relation $\mathfrak{L} = (\frac{\beta}{m}) \cdot \mathfrak{p}_{19} \cdot \mathfrak{p}_{31}^{24}$ where $\beta = 588048307603210005w - 235788727470005542279904$, $m = 19 \cdot 31^{24}$, $\mathfrak{p}_{19} = (19, 2w + 7)$, and $\mathfrak{p}_{31} = (31, 2w + 5)$. Using this factorization, we can then evaluate $\phi: E \rightarrow E'$ using the latter portion of Algorithm 4. We find that E' is

the curve with Weierstrass equation $y^2 = x^3 + 3565469415x + 7170659769$, and $\phi(P) = (7889337683, \pm 3662693258)$. We omit the details of these steps, since this portion of the algorithm is identical to the algorithm of Bröker, Charles and Lauter, and the necessary steps are already extensively detailed in their article [4].

We can check our computations for consistency by performing a second computation, starting from the curve $E' : y^2 = x^3 + 3565469415x + 7170659769$, the point $P' = (7889337683, 3662693258) \in E'(\mathbb{F}_p)$, and the conjugate ideal $\bar{\mathfrak{L}}$, which is represented by the quadratic form $(5000000029, 2326859861, 270713841)$. Let $\bar{\phi} : E' \rightarrow E''$ denote the unique normalized isogeny with kernel $E'[\bar{\mathfrak{L}}]$. Up to a normalization isomorphism $\iota : E \rightarrow E''$, the isogeny $\bar{\phi}$ should equal the dual isogeny $\hat{\phi}$ of ϕ , and the composition $\bar{\phi}(\phi(P))$ should yield $\iota(\ell P)$. Indeed, upon performing the computation, we find that E'' has equation

$$y^2 = x^3 + (15/\ell^4)x + (129/\ell^6),$$

which is isomorphic to E via the isomorphism $\iota : E \rightarrow E''$ defined by $\iota(x, y) = (x/\ell^2, y/\ell^3)$, and

$$\bar{\phi}(\phi(P)) = (3163843645, 8210361642) = (5551543736/\ell^2, 6305164567/\ell^3),$$

in agreement with the value of ℓP , which is $(5551543736, 6305164567)$.

5.2 Medium example

Let E be the ECCp-109 curve [8] from the Certicom ECC Challenge [7], with equation $y^2 = x^3 + ax + b$ over \mathbb{F}_p where

$$\begin{aligned} p &= 564538252084441556247016902735257 \\ a &= 321094768129147601892514872825668 \\ b &= 430782315140218274262276694323197 \end{aligned}$$

As before, to avoid any bias in the choice of ℓ , we set ℓ to be the least Elkies prime greater than $p/2$, and we define $w = \frac{1+\sqrt{\Delta}}{2}$ where $\Delta = \text{disc}(\text{End}(E))$. Let \mathfrak{L} be the prime ideal of norm ℓ in $\text{End}(E)$ corresponding to the reduced quadratic form (ℓ, b, c) of discriminant Δ , where $b = -105137660734123120905310489472471$. For each Elkies prime p , let \mathfrak{p}_p denote the unique prime ideal corresponding to the reduced quadratic form (p, b, c) where $b \geq 0$. Our smoothness bound in this case is $N = L(\frac{1}{2}, \frac{1}{2\sqrt{3}}) \approx 200$. Using Sutherland's `smoothrelation` package [28], which implements the `FINDRELATION` algorithm of [1], one finds in a few seconds (using an initial seed of 0) the relation $\mathfrak{L} = \left(\frac{\beta}{m}\right) \mathfrak{J}$, where

$$\begin{aligned} \mathfrak{J} &= \bar{\mathfrak{p}}_7^{72} \bar{\mathfrak{p}}_{13}^{100} \bar{\mathfrak{p}}_{23}^{14} \bar{\mathfrak{p}}_{47}^2 \bar{\mathfrak{p}}_{73}^2 \bar{\mathfrak{p}}_{103} \mathfrak{p}_{179} \mathfrak{p}_{191} \\ m &= 7^{72} 13^{100} 23^{14} 47^2 73^2 103^1 179^1 191^1 \end{aligned}$$

and

$$\begin{aligned} \beta = & 3383947601020121267815309931891893555677440374614137047492987151 \setminus \\ & 2226041731462264847144426019711849448354422205800884837 \\ & - 1713152334033312180094376774440754045496152167352278262491589014 \setminus \\ & 097167238827239427644476075704890979685 \cdot w \end{aligned}$$

We find that the codomain E' of the normalized isogeny $\phi: E \rightarrow E'$ of kernel $E[\mathfrak{L}]$ has equation $y^2 = x^3 + a'x + b'$ where

$$\begin{aligned} a' &= 84081262962164770032033494307976 \\ b' &= 506928585427238387307510041944828 \end{aligned}$$

and that the base point

$$P = (97339010987059066523156133908935, 149670372846169285760682371978898)$$

of E given in the Certicom ECC challenge has image

$$(450689656718652268803536868496211, \pm 345608697871189839292674734567941).$$

under ϕ . As with the first example, we checked the computation for consistency by using the conjugate ideal.

5.3 Large example

Let E be the ECCp-239 curve [8] from the Certicom ECC Challenge [7]. Then E has equation $y^2 = x^3 + ax + b$ over \mathbb{F}_p where

$$\begin{aligned} p &= 862591559561497151050143615844796924047865589835498401307522524859467869 \\ a &= 820125117492400602839381236756362453725976037283079104527317913759073622 \\ b &= 545482459632327583111433582031095022426858572446976004219654298705912499 \end{aligned}$$

Let \mathfrak{L} be the prime ideal whose norm is the least Elkies prime greater than $p/2$ and whose ideal class is represented by the quadratic form (ℓ, b, c) with $b \geq 0$. We have $N = L(\frac{1}{2}, \frac{1}{2\sqrt{3}}) \approx 5000$, and one finds in a few hours using `smoothrelation` [28] that \mathfrak{L} is equivalent to

$$\mathfrak{J} = \bar{\mathfrak{p}}_7^2 \bar{\mathfrak{p}}_{11} \bar{\mathfrak{p}}_{19} \bar{\mathfrak{p}}_{37}^2 \bar{\mathfrak{p}}_{71}^2 \bar{\mathfrak{p}}_{131} \bar{\mathfrak{p}}_{211} \bar{\mathfrak{p}}_{389} \bar{\mathfrak{p}}_{433} \bar{\mathfrak{p}}_{467} \bar{\mathfrak{p}}_{859}^{18} \bar{\mathfrak{p}}_{863} \bar{\mathfrak{p}}_{1019} \bar{\mathfrak{p}}_{1151} \bar{\mathfrak{p}}_{1597} \bar{\mathfrak{p}}_{2143}^6 \bar{\mathfrak{p}}_{2207}^5 \bar{\mathfrak{p}}_{3359}$$

where each ideal \mathfrak{p}_p is represented by the reduced quadratic form (p, b, c) having $b \geq 0$ (this computation can be reconstructed with [28] using the seed 7). The quotient $\mathfrak{L}/\mathfrak{J}$ is generated by β/m where $m = \text{Norm}(\mathfrak{J})$ and β is

$$\begin{aligned} -923525986803059652225406070265439117913488592374741428959120914067053307 \setminus \\ 4585317 - 917552768623818156695534742084359293432646189962935478129227909w. \end{aligned}$$

Given this relation, evaluating isogenies of degree ℓ is a tedious but routine computation using Elkies-Atkin techniques [4, §3.1]. Although we do not complete it here, the computation is well within the reach of present technology; indeed, Bröker et al. [5] have computed classical modular polynomials mod p of level up to 20000, well beyond the largest prime of 3389 appearing in our relation.

6 Related work

Bisson and Sutherland [1] have developed an algorithm to compute the endomorphism ring of an elliptic curve in subexponential time, using relation-finding techniques which largely overlap with ours. Although our main results were obtained independently, we have incorporated their ideas into our algorithm in several places, resulting in a simpler presentation as well as a large speedup compared to the original version of our work.

Given two elliptic curves E and E' over \mathbb{F}_q admitting a normalized isogeny $\phi: E \rightarrow E'$ of degree ℓ , the equation of ϕ as a rational function contains $O(\ell)$ coefficients. Bostan et al. [3] have published an algorithm which produces this equation, given E , E' , and ℓ . Their algorithm has running time $O(\ell^{1+\varepsilon})$, which is quasi-optimal given the size of the output. Using our algorithm, it is possible to compute E' from E and ℓ in time $\log(\ell)L_{|\Delta|}(\frac{1}{2}, \frac{\sqrt{3}}{2})$ for large ℓ . Hence the combination of the two algorithms can produce the equation of ϕ within a quasi-optimal running time of $O(\ell^{1+\varepsilon})$, given only E and ℓ (or E and \mathfrak{L}), without the need to provide E' in the input.

7 Acknowledgments

We thank the anonymous referees for numerous suggestions which led to substantial improvements in our main result.

References

1. G. Bisson and A. Sutherland. Computing the endomorphism ring of an ordinary elliptic curve over a finite field. *Journal of Number Theory*, to appear, 2009.
2. I. F. Blake, G. Seroussi, and N. P. Smart. *Elliptic curves in cryptography*, volume 265 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 2000. Reprint of the 1999 original.
3. A. Bostan, F. Morain, B. Salvy, and É. Schost. Fast algorithms for computing isogenies between elliptic curves. *Math. Comp.*, 77(263):1755–1778, 2008.
4. R. Bröker, D. Charles, and K. Lauter. Evaluating large degree isogenies and applications to pairing based cryptography. In *Pairing '08: Proceedings of the 2nd international conference on Pairing-Based Cryptography*, pages 100–112, Berlin, Heidelberg, 2008. Springer-Verlag.
5. R. Bröker, K. Lauter, and A. Sutherland. Modular polynomials via isogeny volcanoes, 2010.
6. J. Buchmann and U. Vollmer. *Binary quadratic forms*, volume 20 of *Algorithms and Computation in Mathematics*. Springer, Berlin, 2007. An algorithmic approach.
7. Certicom ECC Challenge. http://www.certicom.com/images/pdfs/cert_ecc_challenge.pdf.
8. Certicom ECC Curves List. <http://www.certicom.com/index.php/curves-list>.
9. H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, and F. Vercauteren, editors. *Handbook of elliptic and hyperelliptic curve cryptography*. Discrete Mathematics and its Applications (Boca Raton). Chapman & Hall/CRC, Boca Raton, FL, 2006.

10. Henri Cohen. *A course in computational algebraic number theory*, volume 138 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, 1993.
11. J.-M. Couveignes and F. Morain. Schoof's algorithm and isogeny cycles. In *Algorithmic number theory (Ithaca, NY, 1994)*, volume 877 of *Lecture Notes in Comput. Sci.*, pages 43–58. Springer, Berlin, 1994.
12. D. A. Cox. *Primes of the form $x^2 + ny^2$* . A Wiley-Interscience Publication. John Wiley & Sons Inc., New York, 1989. Fermat, class field theory and complex multiplication.
13. A. Enge. Computing modular polynomials in quasi-linear time. *Math. Comp.*, 78(267):1809–1824, 2009.
14. M. Fouquet and F. Morain. Isogeny volcanoes and the SEA algorithm. In *Algorithmic number theory (Sydney, 2002)*, volume 2369 of *Lecture Notes in Comput. Sci.*, pages 276–291. Springer, Berlin, 2002.
15. D. Freeman, M. Scott, and E. Teske. A taxonomy of pairing-friendly elliptic curves. *J. Cryptology*, 2010. To appear.
16. S. D. Galbraith. Constructing isogenies between elliptic curves over finite fields. *LMS J. Comput. Math.*, 2:118–138 (electronic), 1999.
17. S. D. Galbraith, F. Hess, and N. P. Smart. Extending the GHS Weil descent attack. In *Advances in cryptology—EUROCRYPT 2002 (Amsterdam)*, volume 2332 of *Lecture Notes in Comput. Sci.*, pages 29–44. Springer, Berlin, 2002.
18. J. Hafner and K. McCurley. A rigorous subexponential algorithm for computation of class groups. *J. Amer. Math. Soc.*, 2(4):837–850, 1989.
19. Kenneth Hardy, Joseph B. Muskat, and Kenneth S. Williams. A deterministic algorithm for solving $n = fu^2 + gv^2$ in coprime integers u and v . *Math. Comp.*, 55(191):327–343, 1990.
20. D. Jao, S. D. Miller, and R. Venkatesan. Do all elliptic curves of the same order have the same difficulty of discrete log? In *Advances in cryptology—ASIACRYPT 2005*, volume 3788 of *Lecture Notes in Comput. Sci.*, pages 21–40. Springer, Berlin, 2005.
21. D. Kohel. *Endomorphism rings of elliptic curves over finite fields*. PhD thesis, University of California, Berkeley, 1996.
22. MAGMA Computational Algebra System. <http://magma.maths.usyd.edu.au/>.
23. A. Menezes, E. Teske, and A. Weng. Weak fields for ECC. In *Topics in cryptology—CT-RSA 2004*, volume 2964 of *Lecture Notes in Comput. Sci.*, pages 366–386. Springer, Berlin, 2004.
24. Arnold Schönhage. Fast reduction and composition of binary quadratic forms. In *ISSAC '91: Proceedings of the 1991 international symposium on Symbolic and algebraic computation*, pages 128–133, New York, NY, USA, 1991. ACM.
25. R. Schoof. Counting points on elliptic curves over finite fields. *J. Théor. Nombres Bordeaux*, 7(1):219–254, 1995. Les Dix-huitièmes Journées Arithmétiques (Bordeaux, 1993).
26. M. Seysen. A probabilistic factorization algorithm with quadratic forms of negative discriminant. *Math. Comp.*, 48(178):757–780, 1987.
27. J. Silverman. *The arithmetic of elliptic curves*, volume 106 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1992. Corrected reprint of the 1986 original.
28. A. Sutherland. smoothrelation. http://math.mit.edu/~drew/smoothrelation_v1.tar.
29. J. Tate. Endomorphisms of abelian varieties over finite fields. *Invent. Math.*, 2:134–144, 1966.
30. E. Teske. An elliptic curve trapdoor system. *J. Cryptology*, 19(1):115–133, 2006.