

# COMPUTING THE ENDOMORPHISM RING OF AN ORDINARY ELLIPTIC CURVE OVER A FINITE FIELD

GAETAN BISSON AND ANDREW V. SUTHERLAND

ABSTRACT. We present two algorithms to compute the endomorphism ring of an ordinary elliptic curve  $E$  defined over a finite field  $\mathbb{F}_q$ . Under suitable heuristic assumptions, both have subexponential complexity. We bound the complexity of the first algorithm in terms of  $\log q$ , while our bound for the second algorithm depends primarily on  $\log |D_E|$ , where  $D_E$  is the discriminant of the order isomorphic to  $\text{End}(E)$ . As a byproduct, our method yields a short certificate that may be used to verify that the endomorphism ring is as claimed.

## 1. INTRODUCTION

Let  $E$  be an ordinary elliptic curve defined over a finite field  $\mathbb{F}_q$ , and let  $\pi$  denote the Frobenius endomorphism of  $E$ . We may view  $\pi$  as an element of norm  $q$  in the integer ring of some imaginary quadratic field  $K = \mathbb{Q}(\sqrt{D_K})$ :

$$(1) \quad \pi = \frac{t + v\sqrt{D_K}}{2} \quad \text{with} \quad 4q = t^2 - v^2 D_K.$$

The trace of  $\pi$  may be computed as  $t = q + 1 - \#E$ . Applying Schoof's algorithm to count the points on  $E/\mathbb{F}_q$ , this can be done in polynomial time [29]. The fundamental discriminant  $D_K$  and the integer  $v$  are then obtained by factoring  $4q - t^2$ , which can be accomplished probabilistically in subexponential time [25].

The endomorphism ring of  $E$  is isomorphic to an order  $\mathcal{O}(E)$  of  $K$ . Once  $v$  and  $D_K$  are known, there are only finitely many possibilities for  $\mathcal{O}(E)$ , since

$$(2) \quad \mathbb{Z}[\pi] \subseteq \mathcal{O}(E) \subseteq \mathcal{O}_K.$$

Here  $\mathbb{Z}[\pi]$  denotes the order generated by  $\pi$ , with discriminant  $D_\pi = v^2 D_K$ , and  $\mathcal{O}_K$  is the maximal order of  $K$  (its ring of integers), with discriminant  $D_K$ . The discriminant of  $\mathcal{O}(E)$  is then of the form  $D_E = u^2 D_K$ , where the conductor  $u$  divides  $v$  and uniquely determines  $\mathcal{O}(E)$ . We wish to compute  $u$ .

Recall that two elliptic curves over  $\mathbb{F}_q$  are isogenous if and only if they have the same trace [20, Ch. 13, Thm. 8.4]. Thus the set  $\text{Ell}_t(\mathbb{F}_q)$  of elliptic curves defined over  $\mathbb{F}_q$  with trace  $t$  constitutes an isogeny class. Each curve in  $\text{Ell}_t(\mathbb{F}_q)$  has an endomorphism ring satisfying (2), and therefore a conductor dividing  $v$ .

In his seminal thesis, Kohel describes the structure of the graph of isogenies defined on  $\text{Ell}_t(\mathbb{F}_q)$ , and its relationship to the orders in  $\mathcal{O}_K$ . He applies this structure to obtain a deterministic algorithm to compute  $u$  in time  $O(q^{1/3+\epsilon})$ , assuming the generalized Riemann hypothesis (GRH) [21, Thm. 24].

Here we present two new methods to compute  $u$  that further exploit the relationship between the isogeny graph and ideal class groups. Under heuristic assumptions (including, but not limited to, the GRH), we achieve subexponential running times. Both methods yield *Las Vegas* algorithms: probabilistic algorithms whose output

is unconditionally correct. We rely on heuristic assumptions only to bound their expected running times.

In practice we find the algorithms perform well, and are able to handle problem sizes that were previously intractable. We give computational examples over finite fields of cryptographic size where  $v$  is large and not smooth (the most difficult case). Over a 200-bit field, for example, the total running time is typically under an hour (see Section 5 for details).

To express our complexity bounds, we adopt the usual notation

$$L[\alpha, c](x) = \exp\left((c + o(1))(\log x)^\alpha (\log \log x)^{1-\alpha}\right).$$

Under the heuristic assumptions detailed in Section 4, we derive the bound

$$L\left[1/2, \sqrt{3}/2\right](q)$$

for Algorithm 1 (Corollary 7), and the bound

$$L[1/2 + o(1), 1](|D_E|) + L[1/3, c_f](q)$$

for Algorithm 2 (Proposition 10). The  $L[1/3, c_f]$  term reflects the heuristic complexity of factoring  $4q - t^2$  using the number field sieve [9]. Algorithm 2 is slower than Algorithm 1 in general, but may be much faster when  $u \ll v$ .

In certain cryptographic applications the discriminant  $D_E$  is an important security parameter (see [6] for one example), and it may be necessary for a third party to independently verify its value. The algorithms we use to compute  $D_E$  may additionally generate a short *certificate* to aid this verification. Both certification and verification have heuristically subexponential running times, and one may extend the certification phase in order to reduce the verification time, as discussed in Section 4. Under the same heuristic assumptions used in our complexity bounds, the size of the certificate is  $O(\log^{2+\epsilon} q)$  (Corollary 8).

## 2. PRELIMINARIES

Kohel's algorithm treats each large prime power  $p^k$  dividing  $v$  by computing the kernel of a certain *smooth isogeny* of degree  $n$ . The prime factors of  $n$  are small (polynomial in  $\log v$ ), but  $n$  itself is large (exponential in  $\log v$ ), and this leads to an exponential running time (see [21, Lem. 29]). We replace this computation with a walk in the isogeny graph using isogenies of low degree (heuristically, subexponential in  $\log v$ ). This walk computes the cardinality of a certain *smooth relation*, and by performing similar computations in class groups of orders in  $\mathcal{O}_K$  we are able to determine the power of  $p$  dividing  $u$  (via Corollary 4). We adapt an algorithm of McCurley [26] to efficiently find smooth relations, achieving a heuristically subexponential running time. First, we present some necessary background.

**2.1. Theoretical background.** Let us fix an ordinary elliptic curve  $E$  defined over a finite field  $\mathbb{F}_q$ , with  $t$ ,  $D_K$ , and  $v$  as in (1). We may verify that  $E$  is ordinary by checking that  $t$  is nonzero modulo the characteristic of  $\mathbb{F}_q$  [34, Prop. 4.31].

Recall that the  $j$ -invariant  $j(E)$  may be computed as a rational function of the coefficients of  $E$  and, in particular, is an element of  $\mathbb{F}_q$ . Over the algebraic closure of  $\mathbb{F}_q$ , the  $j$ -invariant uniquely identifies  $E$  up to isomorphism, but this is not true over  $\mathbb{F}_q$ . However, two ordinary elliptic curves with the same trace are isomorphic over  $\mathbb{F}_q$  if and only if they have the same  $j$ -invariant [12, Prop. 14.19]. Thus we may explicitly represent the set  $\text{Ell}_t(\mathbb{F}_q)$  as a subset of  $\mathbb{F}_q$ , namely, the  $j$ -invariants

of all elliptic curves over  $\mathbb{F}_q$  with trace  $t$ , and view each element of  $\text{Ell}_t(\mathbb{F}_q)$  as a particular elliptic curve representing its isomorphism class.

As noted above, each curve in  $\text{Ell}_t(\mathbb{F}_q)$  has an associated  $u$  dividing  $v$  that identifies its endomorphism ring, and we may partition  $\text{Ell}_t(\mathbb{F}_q)$  into subsets  $\text{Ell}_{t,u}(\mathbb{F}_q)$  accordingly. We aim to distinguish the particular subset containing  $E$  by identifying relations that hold in some  $\text{Ell}_{t,u}(\mathbb{F}_q)$  but not others.

Our main tool is the action of the ideal class group  $\text{cl}(u^2 D_K)$  of  $\mathcal{O}(u^2 D_K)$  (the order of  $K$  with conductor  $u$ ) on the set  $\text{Ell}_{t,u}(\mathbb{F}_q)$ . Here we rely on standard results from the theory of complex multiplication, and the Deuring lifting theorem.

**Theorem 1.** *With  $q, t, v$ , and  $D_K$  as in (1), let  $u$  be a divisor of  $v$  and  $\mathfrak{a}$  an ideal of  $\mathcal{O}(u^2 D_K)$  with prime norm  $\ell$ . Then  $\mathfrak{a}$  acts on the set  $\text{Ell}_{t,u}(\mathbb{F}_q)$  via an isogeny of degree  $\ell$ , and this defines a faithful group action by  $\text{cl}(u^2 D_K)$ .*

For a proof, see Theorems 10.5, 13.12, and 13.14 in [23], or Chapter 3 of [21]. For additional background, we also recommend [12] and [31, Ch. II].

Theorem 1 implies that the cardinality of  $\text{Ell}_{t,u}(\mathbb{F}_q)$  is a multiple of the class number  $h(u^2 D_K)$ , and in fact these values are equal [30]. In general, the curves  $\ell$ -isogenous to  $E$  need not belong to  $\text{Ell}_{t,u}(\mathbb{F}_q)$ . However, when  $\ell$  does not divide  $v$ , we have the following result of Kohel [21, Prop. 23]:

**Theorem 2.** *Let  $\ell$  be a prime not dividing  $v$ . There are exactly  $1 + (D_E | \ell)$  isogenies of degree  $\ell$  starting from  $E$ , and they all lead to curves with endomorphism ring isomorphic to  $\mathcal{O}(E)$ .*

The notation  $(D_E | \ell)$  is the Kronecker symbol. Note that  $(D_E | \ell) = (D_K | \ell)$ , so we can compute it without knowing  $D_E$ . We are primarily interested in the case  $(D_E | \ell) = 1$ , where the prime  $\ell$  splits into distinct prime ideals of norm  $\ell$  in  $\mathcal{O}(E)$ , and these ideals lie in inverse ideal classes  $\alpha$  and  $\alpha^{-1}$  in  $\text{cl}(D_E)$  (if  $\ell$  splits into principal ideals, then  $\alpha = \alpha^{-1} = 1$ ). By Theorem 1, the orbit of  $E$  under the action of  $\alpha$  corresponds to a cycle of  $\ell$ -isogenies whose length is equal to the order of  $\alpha$  in  $\text{cl}(D_E)$ . Additional details on the structure of the isogeny graph can be found in [21] and, in a more concise way, in [15].

**2.2. Explicit computation.** We implement class group computations using binary quadratic forms. For a negative discriminant  $D$ , the ideals in  $\mathcal{O}(D)$  correspond to primitive, positive-definite, binary quadratic forms  $ax^2 + bxy + cy^2$  (commonly noted  $(a, b, c)$ ) with discriminant  $D = b^2 - 4ac$ . The integer  $a$  corresponds to the norm of the ideal. Ideal classes in  $\text{cl}(D)$  are uniquely represented by reduced forms. As typically implemented, the group operation has complexity  $O(\log^2 |D|)$  [5].<sup>1</sup>

To navigate the isogeny graph, we rely on the classical modular polynomial  $\Phi_\ell(X, Y)$ , which parametrizes pairs of  $\ell$ -isogenous elliptic curves. This is a symmetric polynomial with integer coefficients. For a prime  $\ell$  not dividing  $q$ , two elliptic curves  $E_1$  and  $E_2$  defined over  $\mathbb{F}_q$  are connected by an isogeny of degree  $\ell$  if and only if  $\Phi_\ell(j(E_1), j(E_2)) = 0$  [34, Thm. 19].<sup>2</sup>

The polynomial  $\Phi_\ell$  has size  $O(\ell^3 \log \ell)$  [11], and may be computed in time  $O(\ell^{3+\epsilon})$  [14]. When  $\ell$  is small we use precomputed  $\Phi_\ell \in \mathbb{Z}[X, Y]$ , but for larger  $\ell$  we compute  $\Phi_\ell/\mathbb{F}_q$ , that is, the integer polynomial  $\Phi_\ell$  reduced modulo the characteristic of  $\mathbb{F}_q$ . This can be accomplished in time  $O(\ell^{3+\epsilon} \log q)$  and space  $O(\ell^{2+\epsilon} \log q)$  using the

<sup>1</sup>The algorithm of [28] has complexity  $O(\log^{1+\epsilon} |D|)$ , but we do not make use of it.

<sup>2</sup>This isogeny is necessarily cyclic, since it has prime degree.

CRT method described in [7]. In practice one may consider alternative modular polynomials that are sparser and have smaller coefficients than  $\Phi_\ell$ .

To find the curves that are  $\ell$ -isogenous to  $E$ , we compute the roots of the univariate polynomial  $f(X) = \Phi_\ell(X, j(E))$  in  $\mathbb{F}_q$ . We may restrict ourselves to primes  $\ell \nmid v$  with  $(D_E \mid \ell) = 1$ , so that  $f(X)$  has exactly two roots, by Theorem 2. We find these roots by computing  $\gcd(f, X^q - X)$  and solving the resulting quadratic, using an expected  $O(M(\ell) \log q)$  operations in  $\mathbb{F}_q$  (this is the time to compute  $X^q \bmod f$ ). Given  $\Phi_\ell/\mathbb{F}_q$ , we use  $O(\ell^2)$  operations in  $\mathbb{F}_q$  to construct  $f(X) = \Phi_\ell(X, j(E))$ . For  $\ell \gg \log q$  this dominates the time to find the roots of  $f(X)$  and bounds the cost of taking a single step in the  $\ell$ -isogeny graph.

**2.3. Relations.** Let us suppose that  $\alpha \in \text{cl}(D_E)$  contains an ideal of prime norm  $\ell \nmid v$ , and has order  $e = |\alpha|$ . In this situation we say that the relation  $\alpha^e = 1$  holds in  $\text{cl}(D_E)$ . We cannot actually compute  $\alpha^e$  in  $\text{cl}(D_E)$ , since we do not yet know  $D_E$ , but we may apply Theorem 1 to compute the action of either  $\alpha^e$  or  $\alpha^{-e}$  on  $E$  by walking a distance  $e$  along the cycle of  $\ell$ -isogenies, starting from  $j = j(E)$ .

**Algorithm WALKCYCLE( $j, \ell, e$ ):**

1. Set  $j_0 \leftarrow j$ .
2. Let  $j_1$  be one of the two roots of  $\Phi_\ell(X, j_0)$ .
3. For  $s$  from 1 to  $e - 1$ :
4.     Let  $j_{s+1}$  be the root of  $\Phi_\ell(X, j_s)/(X - j_{s-1})$ .
5. Return  $j_e$ .

The roots of  $\Phi_\ell(X, j_s)$  are typically distinct (exceptions require  $|D_E| \leq 4\ell^2$ , by [15, Thm. 2.2]), but the algorithm applies in any case.

The choice of  $j_1$  in Step 2 is arbitrary, it may correspond to the action of either  $\alpha$  or  $\alpha^{-1}$ . Nevertheless, since  $e = |\alpha| = |\alpha^{-1}|$ , we have  $j_e = j_0$  in either case. A difficulty arises when we consider a relation that is not unary, say  $\alpha_1^{e_1} \alpha_2^{e_2} = 1$ , where  $\alpha_i$  contains an ideal of prime norm  $\ell_i$  with  $\ell_1 \neq \ell_2$ . Starting from  $j(E)$ , we walk  $e_1$  steps along the  $\ell_1$ -isogeny cycle, then walk  $e_2$  steps along the  $\ell_2$ -isogeny cycle. We must make two arbitrary choices and may compute the action of  $\alpha_1^{e_1} \alpha_2^{e_2}$ ,  $\alpha_1^{e_1} \alpha_2^{-e_2}$ ,  $\alpha_1^{-e_1} \alpha_2^{e_2}$  or  $\alpha_1^{-e_1} \alpha_2^{-e_2}$ . The actions of these four elements are almost certainly not identical; even if  $\alpha_1^{e_1} \alpha_2^{e_2} = 1$  in  $\text{cl}(D_E)$ , it is unlikely that  $\alpha_1^{e_1} \alpha_2^{-e_2}$  will fix  $j(E)$ .

To address this situation, we formally define a *relation*  $R$  as a pair of vectors  $(\ell_1, \dots, \ell_k)$  and  $(e_1, \dots, e_k)$ , where each  $\ell_i$  is prime,  $\ell_i \nmid v$  and  $(D_K \mid \ell_i) = 1$ , and each  $e_i$  is a positive integer.<sup>3</sup> The integer  $k$  is the *arity* of the relation. Given a discriminant  $D = u^2 D_K$  with  $u \mid v$ , choose ideal classes  $\alpha_1, \dots, \alpha_k \in \text{cl}(D)$  so that  $\alpha_i$  contains an ideal of norm  $\ell_i$ . This ideal need not be the reduced representative of  $\alpha_i$ , and may be principal (implying  $\alpha_i = 1$ ), this depends on  $D$ . We now define

$$(3) \quad \#R/D := \# \left\{ \tau \in \{\pm 1\}^{\{1, \dots, k\}} : \prod_{i=1}^k \alpha_i^{\tau_i e_i} = 1 \right\},$$

as the *cardinality* of the relation  $R$  in  $\text{cl}(D)$ . When  $\#R/D > 0$ , we say  $R$  *holds* in  $\text{cl}(D)$ . The integer  $\#R/D$  is independent of the choice of the  $\alpha_i$ . It has even parity, since if  $\tau$  belongs to the set in (3), so does  $-\tau$ .

To compute  $\#R/D_E$ , we enumerate the  $2^k$  possible walks we may take in the isogeny graph, starting from  $j(E)$ , considering all possible sign vectors  $\tau$  (these

<sup>3</sup>In practice, we may wish to relax the constraint  $\ell_i \nmid v$  when  $\ell_i$  is very small (e.g. 2), see [33].

walks typically form a tree in which each path from root to leaf has  $k$  binary branch points). By the symmetry noted above, we may fix  $\tau_1 = 1$ .

**Algorithm** COUNTRELATION( $E, R$ ):

1. Compute  $j \leftarrow \text{WALKCYCLE}(j(E), \ell_1, e_1)$  and let  $J$  be the list  $(j)$ .
2. For  $i$  from 2 to  $k$ :
3.     Set  $J' \leftarrow J$  and then set  $J$  to the empty list.
4.     For  $j \in J'$ :
5.         Set  $j_0, j'_0 \leftarrow j$  and let  $j_1$  and  $j'_1$  be the two roots of  $\Phi_{\ell_i}(X, j_0)$ .
6.         For  $s$  from 1 to  $e_i - 1$ :
7.             Let  $j_{s+1}$  be the root of  $f(X) = \Phi_{\ell_i}(X, j_s)/(X - j_{s-1})$ .
8.             Let  $j'_{s+1}$  be the root of  $f(X) = \Phi_{\ell_i}(X, j'_s)/(X - j'_{s-1})$ .
9.         Append  $j_{e_i}$  and  $j'_{e_i}$  to  $J$ .
10. Return  $2n$ , where  $n$  counts the occurrences of  $j(E)$  in  $J$ .

Given  $\Phi_{\ell}/\mathbb{F}_q$ , the complexity of Algorithm COUNTRELATION is dominated by

$$(4) \quad \sum_{i=1}^k 2^{i-1} e_i T(\ell_i),$$

where  $T(\ell)$  is the time to take a single step in the  $\ell$ -isogeny graph, which for large  $\ell$  is bounded by  $O(\ell^2)$  operations in  $\mathbb{F}_q$ , as noted above. Our algorithms rely on *smooth* relations in which  $k$ ,  $\ell_i$ , and  $e_i$  are all rather small: in the first example of Section 5 we have  $k = 10$ ,  $\ell_i \leq 500$ , and  $e_i \leq 3000$ . As a practical optimization, we order the couples  $(\ell_i, e_i)$  to minimize (4), using an estimate of  $T(\ell)$ .

Computing  $\#R/D$  in  $\text{cl}(D)$  (where  $D$  is known) is straightforward: one computes the set in (3) by evaluating products of powers in  $\text{cl}(D)$ . A total of  $O(2^k + \sum \log e_i)$  operations in the class group suffice (independent of the  $\ell_i$ ).

**2.4. Probing class groups.** We now consider how we may distinguish class groups of orders in  $K$  by computing the cardinality of suitable relations. We rely on the following lemma.

**Lemma 3.** *Suppose  $\mathcal{O}(D_1) \subseteq \mathcal{O}(D_2)$ . Then for every relation  $R$  we have*

$$\#R/D_1 \leq \#R/D_2.$$

*Proof.* Let  $\mathfrak{a}$  be an  $\mathcal{O}(D_1)$ -ideal with norm prime to the conductor of  $D_1$ . The map

$$\mathfrak{a} \mapsto \mathfrak{a}\mathcal{O}(D_2)$$

induces a natural morphism of class groups. It preserves norms (see [12, Prop. 7.20] for a proof in the case  $D_2$  is fundamental, from which one easily derives the general case) and therefore transports relations from  $\text{cl}(D_1)$  to  $\text{cl}(D_2)$ .  $\square$

**Corollary 4.** *Let  $p^k$  be a prime power dividing  $v$ , and let  $D_1 = (v/p^j)^2 D_K$  and  $D_2 = p^{2k} D_K$ , where  $j = \nu_p(v) - k + 1$ . Suppose  $\#R/D_1 > \#R/D_2$  for some relation  $R$ , and let  $D = u^2 D_K$  where  $u \mid v$ . Then  $p^k \mid u$  if and only if  $\#R/D < \#R/D_1$ .*

Provided we have a suitable relation  $R$  for each prime-power  $p^k$  dividing  $v$ , we can apply the corollary to  $D = D_E$  to determine the prime-power factorization of  $u$ , and hence the endomorphism ring of  $E$ . The computations of  $\#R/D_1$  and  $\#R/D_2$  are performed in the class groups  $\text{cl}(D_1)$  and  $\text{cl}(D_2)$ , but the computation of  $\#R/D_E$  takes place in the isogeny graph via the COUNTRELATION algorithm. Notice that we may replace  $v$  in the corollary by any multiple of  $u$  dividing  $v$ .

**Proposition 5.** *For all primes  $p > 3$ , there are infinitely many relations satisfying the assumptions of Corollary 4.*

*Proof.* Consider unary relations with  $e_1 = 1$  and  $\ell_1 = \ell$ , and denote them  $R_\ell$ . The relation  $R_\ell$  holds in  $\text{cl}(D)$  precisely when  $\ell$  splits into principal ideals in  $\mathcal{O}(D)$ . For  $i \in \{1, 2\}$ , let  $S_i$  be the set of primes  $\ell$  such that  $R_\ell$  holds in  $\text{cl}(D_i)$ . We now show  $S_1 \setminus S_2$  is infinite, referring to material from [12, Ch. 8,9].

The set  $S_i$  is equal to the set of primes that split completely in the ring class field  $L_i$  of  $\mathcal{O}(D_i)$ , and recall that  $L_i$  is a Galois extension of  $\mathbb{Q}$  [12, Lem. 9.3]. The Chebotarev density theorem asserts that  $S_1$  and  $S_2$  are infinite, and  $S_1 \setminus S_2$  is finite if and only if  $L_1 \subseteq L_2$  [12, Thm. 8.19].

But  $L_1$  cannot be contained in  $L_2$ , for  $\mathcal{O}(D_1)$  is not contained in  $\mathcal{O}(D_2)$ . Indeed,  $p^k$  divides the conductor of  $\mathcal{O}(D_2)$  but not that of  $\mathcal{O}(D_1)$ , which implies that  $p^k$  divides the conductor of  $L_2$  but not that of  $L_1$  (see [12, Ex. 9.20–9.23]).  $\square$

In practice, of course, there are many other relations satisfying the requirements of Corollary 4. Empirically, relations  $R$  holding in  $\text{cl}(D_1)$  satisfy  $\#R/D_1 > \#R/D_2$  with probability converging to 1 as  $p$  grows. We will not attempt to prove this statement, but as a heuristic assume that this probability is at least bounded above zero, and furthermore that this applies to relations that are smooth (as defined in Section 4). Note that, independent of this assumption, the above proposition guarantees that our algorithms are always able to terminate.

### 3. ALGORITHMS

**3.1. Computing  $\mathcal{O}(E)$  from above.** We now describe our first algorithm to compute  $u$ , the conductor of the order  $\mathcal{O}(E)$  isomorphic to  $\text{End}(E)$ . We rely on Algorithm `FINDRELATION`( $D_1, D_2$ ), described in Section 3.4, to obtain relations to which Corollary 4 may be applied.

For small primes  $p$  dividing  $v$ , say all  $p \leq B$  for some  $B$ , we can efficiently determine the largest prime power  $p^k$  dividing  $u$  by isogeny climbing, as described in [21, Sec. 4.2] and [33, Sec. 4.1]. This yields an isogenous curve  $E'$  for which the conductor of  $\mathcal{O}(E')$  is  $u' = u/p^k$ , using  $O(kp^2 \log q)$  operations in  $\mathbb{F}_q$  (given  $\Phi_p/\mathbb{F}_q$ ).

For simplicity, the algorithm below assumes that  $v$  is not divisible by the square of a prime larger than  $B$ . The modification to handle large primes whose square divides  $v$  is straightforward but unlikely to be needed in practice.

**Algorithm 1** ( $E/\mathbb{F}_q$ ):

1. Let Schoof's algorithm compute the trace  $t$  of  $E$ , then determine  $D_K, v$ , and the prime factors of  $v$ , by factoring  $4q - t^2 = -v^2 D_K$ .
2. Select a bound  $B$  and set  $u \leftarrow 1$ .
3. For each prime  $p \leq B$  dividing  $v$ :
4.     Determine the largest power of  $p$  dividing  $u$  by isogeny climbing, then set  $E \leftarrow E'$ , remove all powers of  $p$  from  $v$ , and update  $u$ .
5. For each prime  $p > B$  that divides  $v$ :
6.     Set  $R \leftarrow \text{FINDRELATION}(D_1, D_2)$  with  $D_1 = (v^2/p^2)D_K$ ,  $D_2 = p^2 D_K$ .
7.     Determine whether  $p$  divides  $u$  by checking if  $\#R/D_E < \#R/D_1$ , then update  $u$  appropriately.
8. Return  $u$ .

The correctness of Algorithm 1 follows from Corollary 4. Its running time depends on  $B$  and the complexity of FINDRELATION. Using  $B = L[1/2, 1/\sqrt{12}](q)$ , we obtain in Section 4 (Corollary 7) a heuristic bound of

$$L\left[1/2, \sqrt{3}/2\right](q)$$

on the expected running time of Algorithm 1, using  $L[1/2, 1/\sqrt{3}](q)$  space.

Note that the relations computed in Algorithm 1 only depend on the Frobenius trace  $t$  of  $E$ , not its endomorphism ring, hence they may be reused to compute the endomorphism ring of any curve in the same isogeny class. These relations also provide a means to subsequently verify the computation of  $u$ , but for this purpose we may wish to specialize the relations to  $u$ , a task we now consider.

**3.2. Certifying  $u$ .** Let us suppose that a particular value  $u$  is claimed as the conductor of  $\mathcal{O}(E)$ . This may arise in a situation where  $u$  is actually known, either via Algorithm 1 or from the construction of  $E$  (say, by the CM method), but may also occur when one wishes to test a provisional value of  $u$ , as we will do in Algorithm 2. We first give an algorithm to construct a *certificate* that may be used to efficiently check whether a given curve with trace  $t$  in fact has endomorphism ring  $\mathcal{O}(E)$  with conductor  $u$  (equivalently, it allows one to test whether an element of  $\text{Ell}_t(\mathbb{F}_q)$  lies in  $\text{Ell}_{t,u}(\mathbb{F}_q)$ ).

The construction of this certificate depends only on  $u$ ,  $v$ , and  $D_K$  and does not require an elliptic curve as input. Small prime factors of  $u$  and  $v$  may be removed by isogeny climbing prior to calling CERTIFY.

**Algorithm** CERTIFY( $u, v, D_K$ ):

1. For each prime factor  $p$  of  $v/u$ :
2.     Set  $R_p \leftarrow \text{FINDRELATION}(D_1, D_2)$  with  $D_1 = u^2 D_K$  and  $D_2 = p^2 D_K$ .
3. For each prime factor  $p$  of  $u$ :
4.     Set  $R_p \leftarrow \text{FINDRELATION}(D_1, D_2)$  with  $D_1 = (u^2/p^2)D_K$ ,  $D_2 = u^2 D_K$ .
5. Return  $C = (u, v, D_K, \{R_p\}_{p|v})$ .

The relations computed in Step 2 may verify that the actual value of  $u$  divides the claimed value, whereas the relations computed in Step 4 may verify that the claimed value of  $u$  is not a proper divisor of  $u$ , as shown by Algorithm VERIFY.

**Algorithm** VERIFY( $E/\mathbb{F}_q, C$ ):

1. For each prime factor  $p$  of  $v/u$ , verify that  $\#R_p/E > \#R_p/p^2 D_K$ .
2. For each prime factor  $p$  of  $u$ , verify that  $\#R_p/(u^2/p^2)D_K > \#R_p/E$ .
3. Return **true** if all verifications succeed and **false** otherwise.

In addition to the verification of  $u$  above, one may also wish to verify that  $v$  and  $D_K$  are correct. This may be accomplished in polynomial time if the trace  $t$  and the factorizations of  $v$  and  $D_K$  are included in the certificate. One may additionally wish to certify the primes in these factorizations [2], or the verifier may apply a polynomial-time primality test [1]. Assuming these values are correct, the conductor of  $\mathcal{O}(E)$  is equal to  $u$  if and only if VERIFY( $E, C$ ) returns **true**. This statement does not depend on any unproven hypotheses.

The size of the certificate is unconditionally bounded by  $O(\log^3 q)$ , and under heuristic assumptions we obtain an  $O(\log^{2+\epsilon} q)$  bound (Corollary 8). Within this

bound, certificates for primes dividing  $v$  or  $D_K$  can be included, as each certificate requires  $O(\log^{1+\epsilon} q)$  space and there are  $O(\log q)$  such primes.

The expected running times of CERTIFY and VERIFY depend on a smoothness parameter  $\mu$  used by FINDRELATION. This parameter may be chosen to balance the cost of certification and verification, as in Algorithm 2 below, or one may reduce the verification time by increasing the certification time. See Proposition 9 and the discussion following for an analysis of this trade-off.

**3.3. Computing  $\mathcal{O}(E)$  from below.** We now present a second algorithm to compute  $u$ , which may be much faster than Algorithm 1 if  $u$  is small compared to  $v$ , and is in general only slightly slower. Our basic strategy is to examine each of the divisors  $u_i$  of  $v$  in order, attempting to prove that  $u = u_i$  is the conductor of  $\mathcal{O}(E)$ , by constructing a certificate and verifying it. This only requires finding relations in class groups with discriminants whose absolute value is at most  $|u^2 D_k|$ .

Typically  $v$  has few divisors (almost always  $O(\log^{\log^2} v)$  [19, p. 265]), in which case this basic strategy is quite effective. However, in order to improve performance in the worst case, we apply isogeny climbing to effectively remove prime factors from  $v$  as we go, thereby reducing the number of  $u$ 's we must consider. As above, we suppose  $v$  is square-free for the sake of presentation.

**Algorithm 2** ( $E/\mathbb{F}_q$ ):

1. Let Schoof's algorithm compute the trace  $t$  of  $E$ , then determine  $D_K$ ,  $v$ , and the prime factors of  $v$  by factoring  $v^2 D_K = 4q - t^2$ .
2. Set  $x \leftarrow 0$ .
3. Set  $w \leftarrow \max(1/3, x/2 + 1/\log q)$ .
4. For primes  $p < \exp(\log^w v)$ :
5.     Test whether  $p \mid u$  by isogeny climbing, then set  $E \leftarrow E'$  and  $v \leftarrow v/p$ .
6. For divisors  $u$  of  $v$  less than  $\exp(\log^{2w} v)$ :
7.     If VERIFY( $E$ , CERTIFY( $u, v, D_k$ )) returns **true**:  
        Return the product of  $u$  and the primes determined in Step 5.
8. Set  $x \leftarrow 2w$  and go to Step 3.

The variable  $w$  is used to bound the complexity of isogeny climbing using a known lower bound for  $u$  that increases as the algorithm proceeds. Initially we have no information about  $u$  so we use the cost of the factorization computed in Step 1 to select  $w$ .

The running time of Algorithm 2 is analyzed in Section 4, where the bound

$$L[1/2 + o(1), 1] (|D_E|) + L[1/3, c_f](q)$$

is obtained under suitable heuristic assumptions. The same assumptions yield an  $L[1/2 + o(1), 2/3] (|D_E|) \log q$  space bound.

**3.4. Finding Relations.** Given negative discriminants  $D_1$  and  $D_2$ , we seek a relation  $R$  satisfying  $\#R/D_1 > \#R/D_2$ . We find such an  $R$  by searching for a relation that holds in  $\text{cl}(D_1)$  and then testing this inequality. As noted at the end of Section 2, this test almost always succeeds, but if not we search for another relation.

To find relations that hold in  $\text{cl}(D_1)$ , we adapt an algorithm of McCurley [18, 26]. Fix a smoothness bound  $B$ , and for each prime  $\ell \leq B$  with  $(D_1 \mid \ell) \neq -1$ , let  $f_\ell$  denote the *primeform* with norm  $\ell$ . By this we mean the binary quadratic form  $(\ell, b_\ell, c_\ell)$  of discriminant  $D_1$  with  $b_\ell \geq 0$ , which may be constructed via [8, Alg. 3.3].

We then generate random reduced forms by computing the product

$$(5) \quad (a, b, c) = \prod_{\ell} f_{\ell}^{x_{\ell}},$$

where the  $x_{\ell}$  are suitably constrained (and mostly zero). If the prime factors of  $a$  are bounded by  $B$ , say  $a = \prod_{\ell} \ell^{y_{\ell}}$ , then we may decompose  $(a, b, c)$  as

$$(6) \quad (a, b, c) = \prod_{\ell} f_{\ell}^{\tau_{\ell} y_{\ell}},$$

where for nonzero  $y_{\ell}$ ,  $\tau_{\ell} = \pm 1$  is defined by  $b = \tau_{\ell} b_{\ell} \pmod{2\ell}$ .

Recall that  $n = \sqrt{|D_1|/3}$  is an upper bound on the norm of a reduced imaginary quadratic form [13, Ex. 5.14]. Provided that  $\prod_{\ell} \ell^{|x_{\ell}|} > n$ , the decompositions in (5) and (6) must be different, since  $a \leq n$ . This yields a non-trivial relation with exponents  $e_{\ell} = x_{\ell} - \tau_{\ell} y_{\ell}$ .

In order to minimize the cost of computing  $\#R/D_E$  (via `COUNTRELATION`) for the relations we obtain, in addition to bounding the primes  $\ell$ , we must also bound the exponents  $e_{\ell}$ , and especially the number of nonzero  $e_{\ell}$ , which determines the arity  $k$  of  $R$ . To achieve this we require all but a constant number  $k_0$  of the  $x_{\ell}$  to be zero (we use  $k_0 = 3$ ), and note that if we assume  $a$  is a random  $B$ -smooth integer in  $[1, n]$ , then we expect it to have approximately  $2 \log n / \log B$  distinct prime factors. In the unlikely event that  $k$  significantly exceeds this expected value, we seek a different relation.

Having bounded  $k$ , the complexity of `COUNTRELATION` then depends on the products  $e_{\ell} T(\ell)$  appearing in (4). For large  $\ell$  we have  $T(\ell) = O(\ell^2)$  operations in  $\mathbb{F}_q$ . To make the products  $e_{\ell} T(\ell)$  approximately equal we may use the bound  $|x_{\ell}| \leq (B/\ell)^2$ . In practice we use a bound

$$|x_{\ell}| \leq (B/\ell)^{\omega},$$

that better reflects the cost of  $T(\ell)$  for moderate values of  $\ell$  (we typically use  $\omega \approx 1.6$ ); this has no impact on our asymptotic analysis.

The Canfield–Erdős–Pomerance theorem [10, Thm. 3.1] implies that if we sample uniformly random integers from the interval  $[1, n]$  until we find one that is  $L[1/2, \mu](n)$ -smooth, our expected sample size is  $L[1/2, 1/(2\mu)](n)$ , where the implied constants can all be made explicit. This allows us to compute a lower bound  $m(B, n)$  on the number of random integers we must sample from  $[1, n]$  in order to have a better than 50% chance of finding one that is  $B$ -smooth.

We initially set  $B = L[1/2, \mu](n)$ , for a suitably chosen  $\mu$ , and compute  $m(B, n)$  on the heuristic assumption that the norms of the forms we generate are about as likely to be  $B$ -smooth as random integers in the interval  $[1, n]$ .<sup>4</sup> In practice we find this to be the case, however, to account for the possibility that none of the elements generated according to our constraints have  $B$ -smooth norms (or that none of the relations we find are suitable), we increase the smoothness bound by a constant factor  $r$  slightly greater than 1, if we fail to find a suitable relation after testing  $2m(B, n)$  elements.

<sup>4</sup>This is true for random forms, see [8, Prop. 11.4.3].

**Algorithm** FINDRELATION( $D_1, D_2$ ):

1. Set  $B = L[1/2, \mu](n)$ , where  $n = \sqrt{|D_1|/3}$ .
2. Compute primeforms  $f_\ell$  for  $\ell \leq B$ .
3. Repeat  $2m(B, n)$  times:
  4. Pick random integers  $x_\ell$  with  $|x_\ell| \leq (B/\ell)^\omega$  such that at most  $k_0$  of the  $x_\ell$  are nonzero and  $\prod_\ell \ell^{|x_\ell|} > n$ .
  5. Compute the reduced form  $(a, b, c) = \prod_\ell f_\ell^{x_\ell}$ .
  6. If  $a$  is  $B$ -smooth:
    7. Let  $R$  be the relation with  $e_\ell = |x_\ell - \tau_\ell y_\ell|$  where  $a = \prod_\ell \ell^{y_\ell}$ , and let  $k$  be the arity of  $R$ .
    8. If  $k < (2/\mu) \log^{1/2} n$  and  $\#R/D_1 > \#R/D_2$ , then return  $R$ .
  9. Set  $B \leftarrow rB$  and go to Step 2.

As a practical optimization, we may choose not to generate completely new values for  $x_\ell$  every time Step 4 is executed, instead changing just one bit in one of the nonzero  $x_\ell$ . This allows the form  $(a, b, c)$  to be computed in most cases with a single composition/reduction using a precomputed set of binary powers of the  $f_\ell$ .

To implement Step 6 one may use the elliptic curve factorization method (ECM) to probabilistically identify  $B$ -smooth integers in time  $L[1/2, 2](B) = L[1/4, 2\mu](n)$ , which effectively makes the cost of smoothness testing negligible within the precision of our subexponential complexity bounds. A faster approach uses Bernstein's algorithm, which identifies the smooth numbers in a given list in essentially linear time [4]. This does not change our complexity bounds and for the sake of simplicity we use ECM in our analysis.

In practice, the bound  $B$  is quite small (under 1000 in both our examples), and very little time is spent on smoothness testing. In our implementation we used a combination of trial division and a restricted form of Bernstein's algorithm.

#### 4. COMPLEXITY ANALYSIS

The complexity bounds derived below depend on the following heuristics:

- (1) **Small primes.** We assume the GRH. The effective Chebotarev bounds of Lagarias and Odlyzko then imply that for all  $x = \Omega(\log^{2+\epsilon} |D_K|)$  there are  $\Omega(x/\log x)$  primes less than  $x$  that split in  $\mathcal{O}_K$ , where the implied constants are all effectively computable [22, Thm. 1.1].
- (2) **Random norms.** We assume that the norms of the reduced forms computed in Step 4 of FINDRELATION have approximately the distribution of random integers in  $[1, n]$ . Under this assumption, we apply the Canfield–Erdős–Pomerance theorem to estimate the probability of generating a form whose norm is  $B$ -smooth.
- (3) **Random relations.** If  $D_1 = u_1^2 D_K$  and  $D_2 = u_2^2 D_K$  are sufficiently large discriminants with  $u_2 \nmid u_1$ , and  $R$  is a random relation for which  $\#R/D_1 > 0$ , with  $\ell_i$  and  $e_i$  bounded as in FINDRELATION, then we assume that  $\#R/D_1 > \#R/D_2$  with probability bounded above zero.
- (4) **Integer factorization.** We assume that ECM finds a prime factor  $p$  of an integer  $n$  in expected time  $L[1/2, 2](p) \log^2 n$  [24], and that the expected running time of the number field sieve is  $L[1/3, c_f](n)$  [9].

In the propositions and corollaries that follow, we use the shorthand **(H)** to indicate that we are assuming Heuristics 1–4 above.

**Proposition 6.** **(H)**  $\text{FINDRELATION}(D_1, D_2)$  has expected running time

$$L \left[ 1/2, 1/(\sqrt{8}\mu) \right] (|D_1|) + L [1/2, 0] (|D_1|) \log^3 |D_2|.$$

The output relation  $R$  has norms  $\ell_i$  bounded by  $L [1/2, \mu/\sqrt{2}] (|D_1|)$ , exponents  $e_i$  bounded by  $L [1/2, \sqrt{2}\mu] (|D_1|)$ , and arity  $k < (2/\mu) \log^{1/2} |D_1|$ .

*Proof.* Let  $B = L [1/2, \mu] (n) = L [1/2, \mu/\sqrt{2}] (|D_1|)$ , where  $n = \sqrt{|D_1|/3}$ . By Heuristic 1, for sufficiently large  $B$  there are  $\Omega(\log B)$  primes  $\ell = O(\log^2 B)$  with  $(D_1 | \ell) = 1$ . For these  $\ell$ , the value of  $|x_\ell|$  may range up to  $B^{\omega-\delta}$ , for any  $\delta > 0$ . Thus there are more than  $2m(B, n) = L [1/2, 1/(2\mu)]$  distinct elements that may be generated in Step 4, and with high probability at least  $m(B, n)$  are. So Heuristic 2 applies, and with probability greater than 1/2 we generate at least one element with  $B$ -smooth norm each time Step 3 is executed.

Under Heuristic 2, the expected number  $k$  of nonzero exponents  $e_i$  is at most

$$k_0 + 2 \log n / \log B = k_0 + \frac{1}{\mu} \log^{1/2} |D_1| (\log \log |D_1|)^{-1/2},$$

since we expect a random  $B$ -smooth integer in  $[1, n]$  to have  $(2 + o(1)) \log n / \log B$  distinct prime factors (this may be proven with the random bisection model of [3]). This, together with Heuristic 3, ensures that when Step 8 is reached the algorithm terminates, with some constant probability greater than zero. Thus we expect to reach Step 9 just  $O(1)$  times, and the total number of forms  $(a, b, c)$  generated by the algorithm during its execution is bounded by  $L [1/2, 1/(2\mu)] (n)$ .

For each form  $(a, b, c)$ , the algorithm tests whether  $a$  is  $B$ -smooth in Step 6. Applying ECM, under Heuristic 4 we identify a  $B$ -smooth integer (with high probability) in time  $L [1/2, 2] (B) = L [1/4, \sqrt{2}\mu] (n)$  [24]. This yields

$$L [1/2, 1/(2\mu)] (n) = L \left[ 1/2, 1/(\sqrt{8}\mu) \right] (|D_1|),$$

as a bound on the expected time spent finding relations.

The bounds on  $k$ , the  $\ell_i$ , and the  $e_i$  are immediate. We may bound the cost of computing  $\#R/D_j$ , for  $j = \{1, 2\}$ , by

$$O(2^k \log(\max e_i) \log^2 |D_j|) = O(2^k \log^{5/2+\epsilon} |D_j|) = L [1/2, 0] (|D_1|) \log^3 |D_j|.$$

The proposition follows.  $\square$

**Corollary 7.** **(H)** Algorithm 1 has expected running time  $L [1/2, \sqrt{3}/2] (q)$ .

*Proof.* We may compute  $t$  in polynomial time with Schoof's algorithm, and under Heuristic 4 we factor  $4q - t^2 = -v^2 D_K$  in expected time  $L [1/3, c_f] (q)$ .

We use  $B = L [1/2, 1/\sqrt{12}] (q)$  in Algorithm 1, and set  $\mu = 1/\sqrt{6}$  when calling  $\text{FINDRELATION}$ . The cost of isogeny climbing, the calls to  $\text{FINDRELATION}$ , and the calls to  $\text{COUNTRELATION}$  to compute  $\#R/D_E$  all have expected complexity  $L [1/2, \sqrt{3}/2] (q)$ , including the cost of computing the required  $\Phi_\ell/\mathbb{F}_q$ . Only  $O(\log q)$  iterations are required in Algorithm 1 (one for each  $p | v$ ), which does not change the complexity bound.  $\square$

**Corollary 8. (H)** *Let  $D_1 = u^2 D_K$  and  $D_2 = v^2 D_K$ . The expected running time of  $\text{CERTIFY}(u, v, D_K)$  is within an  $O(\log v)$  factor of the expected complexity of  $\text{FINDRELATION}(D_1, D_2)$ . The output certificate  $C$  has size  $O(\log^{1+\epsilon} |D_1| \log v)$ .*

*Proof.* Algorithm  $\text{CERTIFY}$  makes fewer than  $O(\log v)$  calls to  $\text{FindRelation}$  with  $|D_1| \leq |u^2 D_K|$  and  $|D_2| \leq |v^2 D_K|$ . Applying the bounds of Proposition 6 for  $\ell_i$ ,  $e_i$ , and  $k$ , each relation has size  $O(\log |D_1| \log \log |D_1|)$ .  $\square$

**Proposition 9. (H)** *Given a certificate  $C$  produced by Algorithm  $\text{CERTIFY}$  with parameter  $\mu$  and an elliptic curve  $E/\mathbb{F}_q$ , Algorithm  $\text{VERIFY}(E/\mathbb{F}_q, C)$  has expected running time*

$$L \left[ 1/2, 3\mu/\sqrt{2} \right] (|u^2 D_K|) \log^{5/2} q.$$

*Proof.* The expected time to compute  $\Phi_\ell/\mathbb{F}_q$  is  $O(\ell^{3+\epsilon} \log^{1+\epsilon} q)$  [7, 14]. By Proposition 6, each relation in the certificate contains  $O(\log^{1/2} |D_1|)$  distinct  $\ell_i$ , each bounded by  $L \left[ 1/2, \mu/\sqrt{2} \right] (|u^2 D_K|)$ . There are at most  $O(\log q)$  relations in the certificate, yielding a total time of

$$L \left[ 1/2, 3\mu/\sqrt{2} \right] (|u^2 D_K|) \log^{5/2} q,$$

to compute all the  $\Phi_\ell/\mathbb{F}_q$  needed for verification. The total cost of all calls to  $\text{COUNTRELATION}$  may be bounded by

$$L \left[ 1/2, \sqrt{2}\mu \right] (|u^2 D_K|) \log^{2+\epsilon} q,$$

using fast multiplication in  $\mathbb{F}_q$ , which is dominated by the bound above.  $\square$

To balance the costs of verification and certification, one uses  $\mu = 1/\sqrt{6}$ . The verification time may be reduced (and the certification time increased) by choosing a smaller  $\mu$ . For example, with  $\mu = 1/\sqrt{18}$  the verification time is  $L \left[ 1/2, 1/2 \right] (|u^2 D_K|)$  and the certification time is  $L \left[ 1/2, 3/2 \right] (|u^2 D_K|)$ , ignoring logarithmic factors in  $q$ .

**Proposition 10. (H)** *Algorithm 2 has expected running time*

$$L \left[ 1/2 + o(1), 1 \right] (|D_E|) + L \left[ 1/3, c_f \right] (q).$$

*Proof.* In Step 1 we compute  $t$  in polynomial time and factor  $-v^2 D_K$  in expected time  $L \left[ 1/3, c_f \right] (q)$ , by Heuristic 4. Let  $\mu = 1/\sqrt{6}$  in all the calls to  $\text{CERTIFY}$ , in order to balance the cost of  $\text{VERIFY}$ . The cost of each certification/verification performed in Step 7 is then bounded by  $L \left[ 1/2, \sqrt{3}/2 \right] (|D_E|) \log^{5/2} q$ , according to Proposition 9, since we never test a divisor of  $v$  that is greater than the conductor  $u$  of  $D_E$ . In Step 6,  $v$  can contain no prime factors less than  $\exp(\log^w v)$ . Thus the number of divisors is bounded by

$$\left( \frac{\log^{1-w} v}{\log^{2w-w} v} \right) \leq (\log^{1-w} v)^{\log^w v} = L[w, 1](v) = L \left[ 1/2 + o(1), 1 \right] (|D_E|).$$

In the rightmost equality we have used

$$(7) \quad \log |D_E| > \log u \geq \log^{2w-1/\log q} v \quad \Rightarrow \quad \log v \leq \log^{1/(2w-1/\log q)} |D_E|$$

to express the bound in terms of  $|D_E|$ , noting that

$$w/(2w-1/\log q) = 1/2 + 1/(4w \log q - 2),$$

where  $w \geq 1/3$  and  $q \rightarrow \infty$  as  $|D_E| \rightarrow \infty$ . The cost of Step 7 for all the divisors considered in a single execution of Step 6 is bounded by  $L \left[ 1/2 + o(1), 1 \right] (|D_E|) \log^{5/2} q$ .

The algorithm may repeat Step 6 up to  $\log q$  times, but the cost of each iteration dominates all prior ones, so we have bounded the total cost of Step 7.

The cost of isogeny climbing in Step 5 during the first iteration is bounded by

$$\exp\left((3 + o(1)) \log^{1/3} v\right) \log^{2+\epsilon} q = L[1/3, c_f](q)$$

(for any  $c_f$ ), and thereafter cannot exceed

$$\exp((3 + \epsilon) \log^w v) \log^{2+\epsilon} q = L[w, 1](v) \log^3 q = L[1/2 + o(1), 1](|D_E|) \log^{2+\epsilon} q.$$

Here we have again applied (7), and the choice of the constant 1 (or any constant) is justified by the fact that  $3/(\log \log |D_E|)^{1-w} \rightarrow 0$  as  $|D_E| \rightarrow \infty$ .

To complete the proof, we note that if  $L[1/2 + o(1), 1](|D_E|) \log^{5/2} q$  exceeds  $L[1/3, c_f](q)$  we may incorporate the  $\log^{5/2} q$  factor into the  $o(1)$  term. Otherwise, the complexity is  $L[1/3, c_f](q)$ , and the proposition holds in either case.  $\square$

In both Algorithms 1 and 2, the space is dominated by the size of the polynomials  $\Phi_\ell/\mathbb{F}_q$ . As noted in Section 2.2, these can be computed in  $O(\ell^{2+\epsilon} \log q)$  space [7]. Plugging in parameters from the complexity analysis above, and making the same heuristic assumptions, we obtain an  $L[1/2, 1/\sqrt{3}](q)$  space bound for Algorithm 1, and an  $L[1/2 + o(1), 2/3](|D_E|) \log q$  space bound for Algorithm 2.

## 5. EXAMPLES

The rough timings we give here were achieved by a simple implementation running on a single 2.4GHz Intel Q6600 core. The algorithm `FINDRELATION` was implemented using the GNU C/C++ compiler [32] and the GMP library [16], and for `COUNTRELATION` we used a PARI/GP script [27]. We did not attempt to maximize performance, our purpose was simply to demonstrate the practicality of the algorithms on some large inputs. In a more careful implementation, constant factors would be substantially improved and many steps could be parallelized.

**5.1. First Example.** We consider the elliptic curve  $E/\mathbb{F}_q$  with Weierstrass equation  $Y^2 = X^3 - 3X + c_E$ , where

$$\begin{aligned} c_E &= 660897170071025494489036936911196131075522079970680898049528; \\ q &= 160693804425899027550812343206050075546550943415909014478299. \end{aligned}$$

Its trace  $t = 212$  is computed by the Schoof–Elkies–Atkin algorithm in a few seconds and, factoring  $4q - t^2$ , it is nearly instantaneous to retrieve  $D_K = -7$  and

$$v = 2 \cdot 127 \cdot \underbrace{524287}_{p_1} \cdot \underbrace{7195777666870732918103}_{p_2}.$$

Let us apply Algorithm 1 to compute the conductor  $u$  of  $\mathcal{O}(E)$ . First, we use isogeny climbing to handle small prime factors  $p$  of  $v$ , those for which  $\Phi_p$  can be computed in reasonable time (or, more likely, have already been precomputed); here, this means 2 and 127. It takes roughly 20 seconds to compute  $\Phi_{127}$  and isogeny climbing itself takes less than 2 seconds. We find none of these primes divide  $u$ ; hence  $E' = E$  and we may now assume  $v = p_1 p_2$ .

For  $p_1$  we set  $D_1 = (v/p_1)^2 D_K$  and  $D_2 = p_1^2 D_K$  as in Corollary 4. To find a relation satisfying this corollary, we use Algorithm `FINDRELATION`( $D_1, D_2$ ) with the bound  $B = 500$ . Corollary 7 uses  $B = L[1/2, 1/\sqrt{12}](q) \approx 1900$ , but, taking into account constant factors in the complexity estimates, we find experimentally

that  $B = 500$  better balances the expected running time of `FINDRELATION` with that of computing  $\#R/D_E$ . The iteration bound  $2m(B, n) = 6 \cdot 10^7$  has been evaluated via  $m(B, n) = 1/\rho(u)$  with  $u = \log n / \log B \approx 8$  using Table 1 of [17], computed by Bernstein.

After about 20 minutes, `FINDRELATION` outputs the relation  $R$  with

$$(\ell^{e_i}) = (2^{2533}, 11^{752}, 29^2, 37^{47}, 79^1, 113^1, 149^1, 151^2, 347^1, 431^1),$$

for which  $\#R/D_1 = 2$  and  $\#R/D_2 = 0$ . Note that, as suggested by Footnote 3, we make use of  $\ell = 2$  even though it divides  $v$  (using the algorithm in [33, Sec. 4.2]). Now, to evaluate  $\#R/D_E$  using Algorithm `COUNTRELATION`( $E, R$ ), we need to compute the required modular polynomials. We use precomputed  $\Phi_\ell$  for  $\ell < 100$ , and for  $\ell \geq 100$  apply the algorithm in [7];  $\Phi_{431}$  takes 5 minutes,  $\Phi_{347}$  takes 3 minutes, and the others take less than a minute each. Finally,  $\#R/D_E = 0$  is evaluated in 6.5 minutes. Since  $\#R/D_E < \#R/D_1$ , we conclude from Corollary 4 that  $p_1$  is a factor of  $u$ .

We now turn to  $p_2$  and set  $D_1 = (v/p_2)^2 D_K$  and  $D_2 = p_2^2 D_K$  accordingly. The relation  $R = (2^{23}, 11^5, 43^1, 71^2)$  is found almost instantly by `FINDRELATION`( $D_1, D_2$ ), and we have  $\#R/D_1 = 2$  and  $\#R/D_2 = 0$ . `COUNTRELATION`( $E, R$ ) computes  $\#R/D_E = 2$  in 1.5 seconds, proving that  $p_2 \nmid u$  (since  $\#R/D_E \not< \#R/D_1$ ).

All in all, we have found the conductor  $u = p_1$  of the elliptic curve  $E$  defined over a 200-bit prime field in only slightly more than half an hour of computation. The sizes of the primes  $p_1$  and  $p_2$  represents nearly a worst-case; if  $p_2$  was 5 or 6 bits larger the remaining part of  $v$  would be small enough that one could more efficiently use a combination of isogeny climbing and Hilbert class polynomials to determine  $u$ .

We note that, in this example, we could have used the invariant  $\gamma_2 = j^{1/3}$  (or other more favorable invariants [7, 14]) in place of  $j$ , allowing us to use modular polynomials in place of  $\Phi_\ell$  that can be more quickly computed. Doing so would let us increase the bound  $B$  (reducing the time to find relations), and lead to an overall improvement in the running time.

**5.2. Second Example.** Consider now the elliptic curve  $E : Y^2 = X^3 - 3X + c_E$  defined over the 255-bit prime field  $\mathbb{F}_q$  where

$$\begin{aligned} c_E &= 14262957895783764742987524732821199570 \backslash \\ &\quad 860243293007735537575027051453663494306; \\ q &= 50272551883931021408091448710235646749 \backslash \\ &\quad 904660980498576680086699865431843568847. \end{aligned}$$

As above, we compute its trace  $t = 1200$  via the SEA algorithm in about 10 seconds, and an easy factorization yields  $D_K = -7$  and

$$v = 2 \cdot 127 \cdot \underbrace{582509}_{p_1} \cdot \underbrace{582511}_{p_2} \cdot \underbrace{852857}_{p_3} \cdot \underbrace{2305843009213693951}_{p_4}.$$

Let us run Algorithm 2 to compute  $\mathcal{O}(E)$ . We start with  $w = 1/3$ , and first remove the prime factors of  $v$  less than  $\exp(\log^{1/3} v) \approx 85$ . As in Example 1, the constant factors make this a slight underestimate, and we are happy to increase this bound to include both 2 and 127, which we handle by isogeny climbing. We find that neither of these divide  $u$ , and therefore  $E' = E$ , so we now assume that  $v = p_1 p_2 p_3 p_4$ .

We then reach Step 6 and consider divisors  $u$  of  $v$  less than  $\exp(\log^{2w} v) \approx 4 \cdot 10^8$ , namely  $p_1$ ,  $p_2$  and  $p_3$ . Starting with  $u \leftarrow p_1$ , the certificate  $C$  generated in Step 7 by  $\text{CERTIFY}(u, v, D_K)$  consists of

$$R_{p_4} = R_{p_3} = R_{p_2} = (2^{41}, 11^{31}, 37^1) \text{ and } R_{p_1} = (11^1),$$

and takes negligible time to compute. The call to  $\text{VERIFY}(E/\mathbb{F}_q, C)$  takes one second and returns **false**, proving that  $u \neq p_1$ .

Turning to  $u = p_2$ ,  $\text{CERTIFY}(u, v, D_K)$  quickly outputs the certificate

$$R_{p_4} = R_{p_3} = R_{p_1} = (2^{85}, 11^2, 23^5, 29^3) \text{ and } R_{p_2} = (11^1)$$

and  $\text{VERIFY}(E/\mathbb{F}_q, C)$  returns **false** after 1.5 seconds of computation; so  $u \neq p_2$ .

We next consider  $u = p_3$ ; the certificate used is

$$R_{p_4} = R_{p_2} = R_{p_1} = (2^{239}, 11^1, 37^3) \text{ and } R_{p_3} = (11^1).$$

Computing and verifying this certificate takes about a second, and in this case the verification succeeds, proving that  $u = p_3$ .

The total running time is less than 15 seconds, most of which is spent point-counting. For comparison, it takes  $\text{FINDRELATION}(p_2^2 D, p_1^2 D)$  nearly five minutes to output a relation, followed by a twenty-minute computation to evaluate its cardinality, demonstrating the advantage of Algorithm 2 over Algorithm 1 in this example.

#### ACKNOWLEDGMENTS

The authors express their gratitude to Pierrick Gaudry, Takakazu Satoh, Daniel J. Bernstein and Tanja Lange for their careful reading and helpful comments on an early version of this paper.

#### REFERENCES

1. Manindra Agrawal, Neeraj Kayal, and Nitin Saxena, *Primes is in P*, *Annals of Mathematics* **160** (2004), 781–793.
2. A.O.L. Atkin and François Morain, *Elliptic curves and primality proving*, *Mathematics of Computation* **61** (1993), 29–68.
3. Eric Bach, *Analytic methods in the analysis and design of number-theoretic algorithms*, ACM Distinguished Dissertation 1984, MIT Press, 1985.
4. Daniel J. Bernstein, *How to find smooth parts of integers*, 2004, <http://cr.yp.to/papers#smoothparts>.
5. Ingrid Biehl and Johannes Buchmann, *An analysis of the reduction algorithms for binary quadratic forms*, *Voronoi’s Impact on Modern Science* (Peter Engel and Halyna M. Syta, eds.), Institute of Mathematics, Kyiv, 1998, available at <http://www.cdc.informatik.tu-darmstadt.de/reports/TR/TI-97-26.ps.gz>, pp. 71–98.
6. Gaetan Bisson and Takakazu Satoh, *More discriminants with the Brezing-Weng method*, *Progress in Cryptology–INDOCRYPT 2008* (Dipanwita R. Chowdhury, Vincent Rijmen, and Abhijit Das, eds.), *Lecture Notes in Computer Science*, vol. 5365, Springer, 2008, pp. 389–399.
7. Reinier Bröker, Kristen Lauter, and Andrew V. Sutherland, *Computing modular polynomials with the CRT method*, 2009, in preparation.
8. Johannes Buchmann and Ulrich Vollmer, *Binary quadratic forms: An algorithmic approach*, Springer, 2007.
9. Joseph P. Buhler, Hendrik W. Lenstra Jr., and Carl Pomerance, *Factoring integers with the number field sieve*, *The Development of the Number Field Sieve* (Arjen K. Lenstra and Hendrik W. Lenstra Jr., eds.), *Lecture Notes in Mathematics*, vol. 1554, Springer-Verlag, 1993.
10. E. Rodney Canfield, Paul Erdős, and Carl Pomerance, *On a problem of Oppenheim concerning “factorisatio numerorum”*, *Journal of Number Theory* **17** (1983), 1–28.

11. Paula Cohen, *On the coefficients of the transformation polynomials for the elliptic modular function*, Mathematical Proceedings of the Cambridge Philosophical Society **95** (1984), 389–402.
12. David A. Cox, *Primes of the form  $x^2 + ny^2$* , John Wiley & Sons, 1989.
13. Richard Crandall and Carl Pomerance, *Prime numbers: A computational perspective*, second ed., Springer, 2005.
14. Andreas Enge, *Computing modular polynomials in quasi-linear time*, to appear in Mathematics of Computation, arXiv preprint 0704.3177, 2007.
15. Mireille Fouquet and François Morain, *Isogeny volcanoes and the SEA algorithm*, Algorithmic Number Theory Symposium–ANTS V (Claus Fieker and David R. Kohel, eds.), Lecture Notes in Computer Science, vol. 2369, Springer, 2002, pp. 276–291.
16. Torbjörn Granlund et al., *GNU multiple precision arithmetic library*, September 2008, version 4.2.4, available at <http://gmplib.org/>.
17. Andrew Granville, *Smooth numbers: computational number theory and beyond*, Algorithmic Number Theory: Lattices, Number Fields, Curves and Cryptography (Joseph P. Buhler and Peter Stevenhagen, eds.), vol. 44, MSRI Publications, 2008, pp. 267–323.
18. James L. Hafner and Kevin S. McCurley, *A rigorous subexponential algorithm for computing in class groups*, Journal of the American Mathematical Society **2** (1989), no. 4, 837–850.
19. Godfrey H. Hardy and Edward M. Wright, *An introduction to the theory of numbers*, fifth ed., Oxford Science Publications, 1979.
20. Dale Husemöller, *Elliptic curves*, Springer, 1987.
21. David Kohel, *Endomorphism rings of elliptic curves over finite fields*, PhD thesis of the University of California at Berkeley, 1996.
22. Jeffrey C. Lagarias and Andrew M. Odlyzko, *Effective versions of the Chebotarev density theorem*, Algebraic number fields:  $L$ -functions and Galois properties (Proc. Sympos., Univ. Durham, Durham, 1975), Academic Press, 1977, pp. 409–464.
23. Serge Lang, *Elliptic functions*, second ed., Springer, 1987.
24. Hendrik W. Lenstra Jr., *Factoring integers with elliptic curves*, Annals of Mathematics **126** (1987), 649–673.
25. Hendrik W. Lenstra Jr. and Carl Pomerance, *A rigorous time bound for factoring integers*, Journal of the American Mathematical Society **5** (1992), no. 3, 483–516.
26. Kevin S. McCurley, *Cryptographic key distribution and computation in class groups*, NATO ASI on Number Theory and Applications (R.A. Mollin, ed.), C, vol. 265, Kluwer, 1989, pp. 459–479.
27. The PARI group, *PARI/GP*, 2008, version 2.3.4, available at <http://pari.math.u-bordeaux.fr/>.
28. Arnold Schönage, *Fast reduction and composition of binary quadratic forms*, International Symposium on Symbolic and Algebraic Computation–ISSAC’91 (Stephen M. Watt, ed.), ACM Press, 1991, pp. 128–133.
29. René Schoof, *Counting points on elliptic curves over finite fields*, Journal de Théorie des Nombres de Bordeaux **7** (1995), 219–254.
30. Jean-Pierre Serre, *Complex multiplication*, Algebraic Number Theory (John W.S. Cassels and Albrecht Fröhlich, eds.), Academic Press, 1967.
31. Joseph H. Silverman, *Advanced topics in the arithmetic of elliptic curves*, Springer, 1999.
32. Richard Stallman et al., *GNU compiler collection*, August 2008, version 4.3.2, available at <http://gcc.gnu.org/>.
33. Andrew V. Sutherland, *Computing Hilbert class polynomials with the Chinese Remainder Theorem*, 2009, <http://arxiv.org/abs/0903.2785>.
34. Lawrence C. Washington, *Elliptic curves: Number theory and cryptography*, second ed., CRC Press, 2008.