

## MODULAR POLYNOMIALS VIA ISOGENY VOLCANOES

REINIER BRÖKER, KRISTIN LAUTER, AND ANDREW V. SUTHERLAND

ABSTRACT. We present a new algorithm to compute the classical modular polynomial  $\Phi_l$  in the rings  $\mathbf{Z}[X, Y]$  and  $(\mathbf{Z}/m\mathbf{Z})[X, Y]$ , for a prime  $l$  and any positive integer  $m$ . Our approach uses the graph of  $l$ -isogenies to efficiently compute  $\Phi_l \bmod p$  for many primes  $p$  of a suitable form, and then applies the Chinese Remainder Theorem (CRT). Under the Generalized Riemann Hypothesis (GRH), we achieve an expected running time of  $O(l^3(\log l)^3 \log \log l)$ , and compute  $\Phi_l \bmod m$  using  $O(l^2(\log l)^2 + l^2 \log m)$  space. We have used the new algorithm to compute  $\Phi_l$  with  $l$  over 5000, and  $\Phi_l \bmod m$  with  $l$  over 20000. We also consider several modular functions  $g$  for which  $\Phi_l^g$  is smaller than  $\Phi_l$ , allowing us to handle  $l$  over 60000.

### 1. INTRODUCTION

For a prime  $l$ , the classical modular polynomial  $\Phi_l$  is the minimal polynomial of the function  $j(lz)$  over the field  $\mathbf{C}(j)$ , where  $j(z)$  is the modular  $j$ -function. The polynomial  $\Phi_l$  parametrizes elliptic curves  $E$  together with an isogeny  $E \rightarrow E'$  of degree  $l$ . From classical results, we know that  $\Phi_l$  lies in the ring  $\mathbf{Z}[X, Y]$  and satisfies  $\Phi_l(X, Y) = \Phi_l(Y, X)$ , with degree  $l + 1$  in both variables [58, §69].

The fact that the moduli interpretation of  $\Phi_l$  remains valid modulo primes  $p \neq l$  was crucial to the improvements made by Atkin and Elkies to Schoof's point-counting algorithm [22, 51]. More recently, the polynomials  $\Phi_l \bmod p$  have been used to compute Hilbert class polynomials [2, 54], and to determine the endomorphism ring of an elliptic curve over a finite field [6]. Explicitly computing  $\Phi_l$  is notoriously difficult, primarily due to its large size. As shown in [20], the logarithmic height of its largest coefficient is  $6l \log l + O(l)$ , thus its total size is

$$(1) \quad O(l^3 \log l).$$

As this bound suggests, the size of  $\Phi_l$  grows quite rapidly; the binary representation of  $\Phi_{79}$  already exceeds one megabyte, and  $\Phi_{659}$  is larger than a gigabyte.

The polynomial  $\Phi_l$  can be computed by comparing coefficients in the Fourier expansions of  $j(z)$  and  $j(lz)$ , an approach considered by several authors [8, 22, 37, 39, 41, 45, 47]. As detailed in [8], this only requires integer arithmetic, and may be performed modulo  $p$  for any prime  $p > 2l + 2$ . The time to compute  $\Phi_l \bmod p$  is then  $O(l^{3+\varepsilon}(\log p)^{1+\varepsilon})$ , and for a sufficiently large  $p$  this yields an  $O(l^{4+\varepsilon})$  time algorithm to compute  $\Phi_l$  over  $\mathbf{Z}$ . Alternatively (and preferably), one computes  $\Phi_l$  modulo several smaller primes and applies the Chinese Remainder Theorem, as suggested in [8, 37, 45, 47].

An alternative CRT-based approach appears in [17]. This algorithm uses isogenies between supersingular elliptic curves defined over a finite field, and computes  $\Phi_l \bmod p$  in time  $O(l^{4+\varepsilon}(\log p)^{2+\varepsilon} + (\log p)^{4+\varepsilon})$ , under the GRH.

---

2000 *Mathematics Subject Classification.* Primary 11Y16 ; Secondary 11G15, 11G20, 14H52.

In [24], Enge uses interpolation and fast floating-point evaluations to compute  $\Phi_l \in \mathbf{Z}[X, Y]$  in time  $O(l^3(\log l)^{4+\varepsilon})$ , under reasonable heuristic assumptions. The complexity of this method is nearly optimal, quasi-linear in the size of  $\Phi_l$ . However, most applications actually use  $\Phi_l$  in a finite field  $\mathbf{F}_{p^n}$ , and  $\Phi_l \bmod p$  may be much smaller than  $\Phi_l$ . In general, Enge’s algorithm can compute  $\Phi_l$  and reduce it modulo  $p$  much faster than either of the methods above can compute  $\Phi_l \bmod p$ , but this may use an excessive amount of space. For large  $l$  this approach becomes impractical, even when  $\Phi_l \bmod p$  is reasonably small.

Here we present a new method to compute  $\Phi_l$ , either over the integers or modulo an arbitrary positive integer  $m$ , including  $m \leq l$ . Our algorithm is both asymptotically and practically faster than alternative methods, and achieves essentially optimal space complexity. More precisely, we prove the following result.

**Theorem 1.** *Let  $l$  denote an odd prime and  $m$  a positive integer. Algorithm 6.1 correctly computes  $\Phi_l \in (\mathbf{Z}/m\mathbf{Z})[X, Y]$ . Under the GRH, it runs in expected time*

$$O(l^3 \log^3 l \log \log l),$$

*using  $O(l^2 \log lm)$  expected space.*

To compute  $\Phi_l$  over  $\mathbf{Z}$ , we choose a modulus  $m$  that is large enough to uniquely determine the coefficients, via an explicit height bound proven in [13]. In general, we may assume  $\log m = O(l \log l)$ , since otherwise  $\Phi_l$  and  $\Phi_l \bmod m$  are effectively the same (hence the time bound does not depend on  $m$ ).

Our algorithm is of the *Las Vegas* type, a probabilistic algorithm whose output is unconditionally correct; the GRH is only used to analyze its running time. We have used it to compute  $\Phi_l$  for all  $l < 3600$ , and many larger  $l$  up to 5003. The largest previous computation of which we are aware has  $l = 1009$ . Working modulo  $m$  we can go further; we have computed  $\Phi_l$  modulo a 256-bit integer  $m$  with  $l = 20011$ .

Applications that rely on  $\Phi_l$  can often improve their running times by using alternative modular polynomials that have smaller coefficients. Our algorithm can be adapted to compute polynomials  $\Phi_l^g$  relating  $g(z)$  and  $g(lz)$ , for modular functions  $g$  that share certain properties with  $j$ . This includes the cube root  $\gamma_2$  of  $j$ , and we are then able to compute  $\Phi_l \bmod m$  more quickly by reconstructing it from  $\Phi_l^{\gamma_2} \bmod m$ , capitalizing on a suggestion in [22]. Other examples include simple and double eta-quotients, the Atkin functions, and the Weber  $\mathfrak{f}$ -function. The last is especially attractive, since the modular polynomials for  $\mathfrak{f}$  are approximately 1728 times smaller than those for  $j$ . This has allowed us to compute modular polynomials  $\Phi_l^{\mathfrak{f}}$  with  $l$  as large as 60013.

The outline of this article is as follows. In Section 2 we give a rough overview of our new algorithm. The theory behind the algorithm is presented in Sections 3–5. We present the algorithm, prove its correctness and analyze its runtime in Section 6. Section 7 deals with modular polynomials for modular functions other than  $j$ , and a final Section 8 contains computational results.

## 2. OVERVIEW

Our basic strategy is a standard CRT approach: we compute  $\Phi_l \bmod p$  for various primes  $p$  and use the Chinese Remainder Theorem to recover  $\Phi_l \in \mathbf{Z}[X, Y]$ . Alternatively, the explicit CRT (mod  $m$ ) allows us to directly compute  $\Phi_l \in (\mathbf{Z}/m\mathbf{Z})[X, Y]$ , via [4, Thm. 3.1]. By applying the algorithm of [54, §6], this can be accomplished in  $O(l^2 \log lm)$  space, even though the total size of all the  $\Phi_l \bmod p$  is  $O(l^3 \log l)$ .

Our method for computing  $\Phi_l \bmod p$  is new, and applies only to certain primes  $p$ . Strategic prime selection has been used effectively in other CRT-based algorithms, such as [54], and it is especially helpful here. Working in the finite field  $\mathbf{F}_p$ , we select  $l + 2$  distinct values  $j_i$ , compute  $\Phi_l(X, j_i) \in \mathbf{F}_p[X]$  for each, and then interpolate the coefficients of  $\Phi_l \in (\mathbf{F}_p[Y])[X]$  as polynomials in  $\mathbf{F}_p[Y]$ . The key lies in our choice of  $p$ , which allows us to select particular interpolation points that greatly facilitate the computation. We are then able to compute  $\Phi_l \bmod p$  in expected time

$$(2) \quad O(l^2(\log p)^3 \log \log p).$$

In contrast to the methods above, this is quasi-linear in the size of  $\Phi_l \bmod p$ .

Our algorithm exploits the structure of the  $l$ -isogeny graph  $G_l$  defined on the set of  $j$ -invariants of elliptic curves over  $\mathbf{F}_p$ . Each edge in this graph corresponds to an  $l$ -isogeny; the edge  $(j_1, j_2)$  is present if and only if  $\Phi_l(j_1, j_2) = 0$ . As described in [30, 42], the ordinary components of this graph have a particular structure known as an  $l$ -volcano. Depicted in Figure 1 are a set of four  $l$ -volcanoes, each with two levels: the *surface* (at the top), and the *floor* (on the bottom). Note that each vertex  $j_i$  on the surface has  $l + 1$  neighbors, these are the roots of  $\Phi_l(X, j_i) \in \mathbf{F}_p[X]$ , and there are at least  $l + 2$  such  $j_i$ .

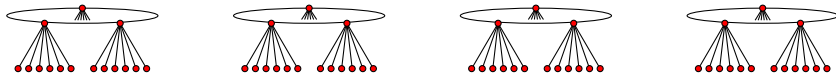


FIGURE 1. A set of  $l$ -volcanoes arising from Theorem 4.1. In this example  $l = 7$  splits into ideals of order 3 in  $\text{cl}(\mathcal{O})$  and we have  $h(\mathcal{O}) = 12$  surface curves and  $h(R) = 72$  floor curves.

This configuration contains enough information to compute the  $l + 2$  polynomials  $\Phi_l(X, j_i)$  that we need to interpolate  $\Phi_l(X, Y) \bmod p$ . It is not an arrangement that is likely to arise by chance; it is achieved by our choice of the order  $\mathcal{O}$  and the primes  $p$  that we use. To further simplify our task, we choose  $p$  so that vertices on the surface correspond to curves with  $\mathbf{F}_p$ -rational  $l$ -torsion. Our ability to obtain such primes is guaranteed by Theorems 4.1 and 4.4, proven in Section 4.

The curves on the surface all have the same endomorphism ring type, isomorphic to an imaginary quadratic order  $\mathcal{O}$ . Their  $j$ -invariants are precisely the roots of the Hilbert class polynomial  $H_{\mathcal{O}} \in \mathbf{Z}[X]$ . As described in [2], the roots of  $H_{\mathcal{O}}$  may be enumerated via the action of the ideal class group  $\text{cl}(\mathcal{O})$ . To do so efficiently, we use an algorithm of [54] to compute a *polycyclic presentation* for  $\text{cl}(\mathcal{O})$  that allows us to enumerate the roots of  $H_{\mathcal{O}}$  via isogenies of low degree, typically much smaller than  $l$ . We may use this presentation to determine the action of any element of  $\text{cl}(\mathcal{O})$ , including those that act via  $l$ -isogenies. This allows us to identify the  $l$ -isogeny cycles that form the surfaces of the volcanoes in Figure 1.

Similarly, the vertices on the floor are the roots of  $H_R$ , where  $R$  is the order of index  $l$  in  $\mathcal{O}$ , and we use a polycyclic presentation of  $\text{cl}(R)$  to enumerate them. To identify children of a common parent (siblings), we exploit the fact that siblings lie in a cycle of  $l^2$ -isogenies, which we identify using our presentation of  $\text{cl}(R)$ . It remains only to connect each parent to one of its children. This may be achieved by using Vélú's formula [55] to compute an  $l$ -isogeny from the surface to the floor. By matching each parent to a group of siblings, we avoid the need to compute an  $l$ -isogeny to every child, which is critical to obtaining the complexity bound in (2).

Below is a simplified version of the algorithm to compute  $\Phi_l \bmod p$ .

**Algorithm 2.1.** Let  $l$  be an odd prime, and let  $\mathcal{O}$  be an imaginary quadratic order of discriminant  $D$  with class number  $h(\mathcal{O}) \geq l + 2$ . Let  $p \equiv 1 \pmod l$  be a prime satisfying  $4p = t^2 - l^2v^2D$  for some integers  $t$  and  $v$  with  $l \nmid v$ . Let  $R$  be the order of index  $l$  in  $\mathcal{O}$ . Compute  $\Phi_l \bmod p$  as follows:

1. Find a root of  $H_{\mathcal{O}}$  over  $\mathbf{F}_p$ .
2. Enumerate the roots  $j_i$  of  $H_{\mathcal{O}}$  and identify the  $l$ -isogeny cycles.
3. For each  $j_i$  find an  $l$ -isogenous  $j$  on the floor.
4. Enumerate the roots of  $H_R$  and identify the  $l^2$ -isogeny cycles.
5. For each  $j_i$  compute  $\Phi_l(X, j_i) = \prod_{(j_i, j_k) \in G_l} (X - j_k)$ .
6. Interpolate  $\Phi_l \in (\mathbf{F}_p[Y])[X]$  using the  $j_i$  and the polynomials  $\Phi_l(X, j_i)$ .

The conditions on the inputs  $l$ ,  $\mathcal{O}$ , and  $p$  suffice to ensure that Theorem 4.1 is satisfied, so that we have a configuration of  $l$ -volcanoes similar to the example in Figure 1. We use the same  $\mathcal{O}$  for each  $p$ , so the Hilbert class polynomial  $H_{\mathcal{O}}$  may be precomputed, but we do not need to compute  $H_R$ , instead we enumerate its roots by applying the Galois action of  $\text{cl}(R)$  to a root obtained in Step 3.

A more detailed version of Algorithm 2.1 appears in Section 6 together with Algorithm 6.1, which selects the order  $\mathcal{O}$  and the primes  $p$ , and performs the CRT computations needed to determine  $\Phi_l$  over  $\mathbf{Z}$ , or modulo  $m$ .

### 3. ORDERS IN IMAGINARY QUADRATIC FIELDS

It is a classical fact that the endomorphism ring of an ordinary elliptic curve over a finite field is isomorphic to an imaginary quadratic order  $\mathcal{O}$ . The order  $\mathcal{O}$  is necessarily contained in the maximal order  $\mathcal{O}_K$  of its fraction field  $K$ , but we quite often have  $\mathcal{O} \subsetneq \mathcal{O}_K$ . As most textbooks on algebraic number theory focus on maximal orders, we first develop some useful tools for working with non-maximal orders. To simplify the presentation, we work throughout with fields of discriminant  $d_K < -4$ , ensuring that we always have the unit groups  $\mathcal{O}^* = \mathcal{O}_K^* = \{\pm 1\}$ . We use  $\left(\frac{d_K}{p}\right)$  to denote the Kronecker symbol, which is  $-1, 0$ , or  $1$  as the prime  $p$  splits, ramifies, or remains inert in  $K$  (respectively).

Let  $\mathcal{O}$  be a (not necessarily maximal) order in a quadratic field  $K$  of discriminant  $d_K < -4$ . Let  $N$  be a positive integer prime to the conductor  $u = [\mathcal{O}_K : \mathcal{O}]$ . The order  $R = \mathbf{Z} + N\mathcal{O}$  has index  $N$  in  $\mathcal{O}$ , and its ideal class group  $\text{cl}(R)$  is an extension of  $\text{cl}(\mathcal{O})$ . More precisely, as in [53, Thm. 6.7], there is an exact sequence

$$(3) \quad 1 \longrightarrow (\mathcal{O}/N\mathcal{O})^*/(\mathbf{Z}/N\mathbf{Z})^* \longrightarrow \text{cl}(R) \xrightarrow{\varphi} \text{cl}(\mathcal{O}) \longrightarrow 1,$$

where  $\varphi$  maps the class  $[I]$  to the class  $[I\mathcal{O}]$ . For  $R$ -ideals prime to  $uN$ , the underlying map  $I \mapsto I\mathcal{O}$  preserves ‘norms’, that is,  $[R : I] = [\mathcal{O} : I\mathcal{O}]$ , as in [21, Prop. 7.20]. We have a particular interest in the kernel of the map  $\varphi$ .

**Lemma 3.1.** *In the exact sequence above, if  $N = p^n$  is a power of an unramified odd prime  $p$ , then  $\ker \varphi$  is cyclic of order  $p^{n-1}(p - \left(\frac{d_K}{p}\right))$ .*

*Proof.* We compute the structure of  $\ker \varphi \cong (\mathcal{O}/p^n\mathcal{O})^*/(\mathbf{Z}/p^n\mathbf{Z})^*$ . The group  $(\mathbf{Z}/p^n\mathbf{Z})^*$  is cyclic, isomorphic to the additive group  $(\mathbf{Z}/(p-1)\mathbf{Z}) \times (\mathbf{Z}/p^{n-1}\mathbf{Z})$ . We now apply [18, Cor. 4.2.11] to compute the structure of  $(\mathcal{O}/p^n\mathcal{O})^*$ :

$$(4) \quad (\mathcal{O}/p^n\mathcal{O})^* \cong \begin{cases} (\mathbf{Z}/(p-1)\mathbf{Z})^2 \times (\mathbf{Z}/p^{n-1}\mathbf{Z})^2, & \text{if } p \text{ splits in } K; \\ (\mathbf{Z}/(p^2-1)\mathbf{Z}) \times (\mathbf{Z}/p^{n-1}\mathbf{Z})^2, & \text{if } p \text{ is inert in } K. \end{cases}$$

In both cases, the factor  $(\mathbf{Z}/p^{n-1}\mathbf{Z})$  of  $(\mathbf{Z}/p^n\mathbf{Z})^*$  is a maximal cyclic subgroup of the Sylow  $p$ -subgroup of  $(\mathcal{O}/p^n\mathcal{O})^*$ , and must correspond to a direct summand. Thus the  $p$ -rank of the quotient  $(\mathcal{O}/p^n\mathcal{O})^*/(\mathbf{Z}/p^n\mathbf{Z})^*$  is 1. The order of the factor  $(\mathbf{Z}/(p-1)\mathbf{Z})$  of  $(\mathbf{Z}/p^n\mathbf{Z})^*$  is not divisible by  $p$ , and must correspond to a subgroup of a cyclic factor of  $(\mathcal{O}/p^n\mathcal{O})^*$  in both cases. It follows that the quotient is cyclic. The calculations above also show that  $\#(\mathcal{O}/p^n\mathcal{O})^*/(\mathbf{Z}/p^n\mathbf{Z})^* = p^{n-1}(p - \binom{d_K}{p})$ .  $\square$

Even when  $\ker \varphi$  is not necessarily cyclic, the size of  $\ker \varphi$  is as in Lemma 3.1. More generally, the exact sequence (3) can be used to derive the formula

$$(5) \quad h(\mathcal{O}) = h(\mathcal{O}_K)u \prod_{p|u} \left( 1 - \left( \frac{d_K}{p} \right) p^{-1} \right),$$

as in [21, Thm. 7.24].

We now describe a particular representation of  $\ker \varphi$  when  $N = l$  is prime. In this case  $\ker \varphi$  is cyclic, of order  $l - \binom{d_K}{l}$ ; this follows from Lemma 3.1 for  $l > 2$ , and from (5) for  $l = 2$ . Let  $\mathcal{O} = \mathbf{Z}[\tau]$  for some  $\tau \in K$  that is coprime to  $l$ . There are exactly  $l + 1$  index  $l$  sublattices of  $\mathcal{O}$ : the order  $R$ , and lattices  $S_i = l\mathbf{Z} + (\tau + i)\mathbf{Z}$ , for  $i$  from 0 to  $l - 1$ . Each  $\mathcal{O}$ -ideal of norm  $l$  corresponds to one of the  $S_i$ . The remaining  $S_i$  are fractional invertible  $R$ -ideals corresponding to proper  $R$ -ideals

$$(6) \quad J_i = lS_i = l^2\mathbf{Z} + l(\tau + i)\mathbf{Z},$$

for which  $R = \{\beta \in K : \beta J_i \subset J_i\}$ . Exactly  $1 + \binom{d_K}{l}$  of the  $S_i$  are  $\mathcal{O}$ -ideals, leaving  $l - 1 - \binom{d_K}{l}$  proper  $R$ -ideals  $J_i$ . These are all non-principal and inequivalent in  $\text{cl}(R)$ , and each lies in  $\ker \varphi$ , since we have  $J_i\mathcal{O} = l\mathcal{O}$ . The invertible  $J_i$  are exactly the non-trivial elements of  $\ker \varphi$ . We summarize with the following lemma, which guarantees that we can find a generator for the cyclic group  $\ker \varphi$  that has norm  $l^2$ .

**Lemma 3.2.** *If  $N = l$  is prime in the exact sequence (3), then the  $R$ -ideal  $lR$  and the invertible  $R$ -ideals  $J_i$  defined in (6) are representatives for  $\ker \varphi$ . In particular,  $\ker \varphi$  is generated by the class of an invertible  $R$ -ideal with norm  $l^2$ .*

This representation of  $\ker \varphi$  has proven useful in other settings [16]. We use it to obtain the  $l^2$ -isogeny cycles we need in Step 4 of Algorithm 2.1.

We conclude this section with a theorem that allows us to construct arbitrarily large class groups that are generated by elements of bounded norm.

**Theorem 3.3.** *Let  $\mathcal{O}$  be an order in a quadratic field of discriminant  $d_K < -4$ , and let  $p \nmid \text{disc}(\mathcal{O})$  be an odd prime. Let  $\mathcal{P}$  be a set of primes that do not divide  $p[\mathcal{O}_K : \mathcal{O}]$ . For  $n \in \mathbf{Z}_{\geq 0}$ , let  $R_n$  denote the order  $\mathbf{Z} + p^n\mathcal{O}$ , and let  $G_n$  be the subgroup of  $\text{cl}(R_n)$  generated by the set  $S_n$  of classes of  $R_n$ -ideals with norms in  $\mathcal{P}$ .*

*Then if  $G_2 = \text{cl}(R_2)$ , we have  $G_n = \text{cl}(R_n)$  for every  $n \in \mathbf{Z}_{\geq 0}$ .*

*Proof.* For each  $R_n$ , let  $\varphi_n : \text{cl}(R_n) \rightarrow \text{cl}(\mathcal{O})$  denote the corresponding map in the exact sequence (3), and let  $\phi_{n+1} : \text{cl}(R_{n+1}) \rightarrow \text{cl}(R_n)$  send  $[I]$  to  $[IR_n]$ , so that  $\varphi_{n+1} = \varphi_n \circ \phi_{n+1}$ . These are all surjective group homomorphisms, and the underlying ideal maps preserve the norms of ideals prime to  $p[\mathcal{O}_K : \mathcal{O}]$ . We assume  $G_2 = \text{cl}(R_2)$ , which implies  $G_n = \text{cl}(R_n)$  for  $n \leq 2$ , and proceed by induction on  $n$ .

For each prime  $q \in \mathcal{P}$  and every  $n$ , there are exactly  $1 - \left(\frac{d_K}{q}\right)$  ideals in  $R_n$  of norm  $q$ , and  $\phi_{n+1}$  maps  $S_{n+1}$  onto  $S_n$  and  $G_{n+1}$  onto  $G_n$ . By the inductive hypothesis,  $G_n = \text{cl}(R_n)$ , therefore  $G_{n+1}$  intersects every coset of  $\ker \phi_{n+1} \subset \ker \varphi_{n+1}$ . To prove  $G_{n+1} = \text{cl}(R_{n+1})$ , it suffices to show  $\ker \varphi_{n+1} \subset G_{n+1}$ .

The groups  $\ker \varphi_n$  and  $\ker \varphi_{n+1}$  are cyclic, by Lemma 3.1, since  $p$  is odd and unramified. Let  $\alpha_n$  be a generator for  $\ker \varphi_n$ . Since  $\#\ker \varphi_n$  is divisible by  $p$ ,  $\alpha_n$  cannot be a  $p$ th power in  $\ker \varphi_n$ . Expressing  $\alpha_n$  in terms of  $S_n$ , we see that  $\phi_{n+1}^{-1}(\alpha_n)$  must intersect  $G_{n+1}$ . Let  $\alpha_{n+1}$  lie in this intersection, and note that  $\alpha_{n+1} \in \ker \varphi_{n+1}$ . The order of  $\alpha_{n+1}$  must be a multiple of  $|\alpha_n| = \#\ker \varphi_n$ , and  $\alpha_{n+1}$  cannot be a  $p$ th power in  $\ker \varphi_{n+1}$ . It follows that  $\alpha_{n+1}$  has order  $\#\ker \varphi_{n+1}$ , hence it generates  $\ker \varphi_{n+1}$ , proving  $\ker \varphi_{n+1} \subset G_{n+1}$  as desired.  $\square$

To see Theorem 3.3 in action, let  $\mathcal{O}$  be the order of discriminant  $D = -7$ , let  $p = 3$ , and let  $\mathcal{P} = \{2\}$ . The class group of the order  $R_n$  of discriminant  $3^{2n}D$  happens to be generated by an ideal of norm 2 when  $n = 2$ , and the theorem then implies that this holds for all  $n$ . This allows us to construct arbitrarily large cyclic class groups, each generated by an ideal of norm 2.

We remark that Theorem 3.3 may be extended to handle  $p = 2$  if the condition  $G_2 = \text{cl}(R_2)$  is replaced by  $G_3 = \text{cl}(R_3)$ , and easily generalizes to treat families of orders lying in  $\mathcal{O}$  that have  $b$ -smooth conductors, for any constant  $b$ .

#### 4. EXPLICIT CM THEORY

**4.1. The theory of complex multiplication (CM).** As in Section 3, let  $\mathcal{O}$  be an order in a quadratic field  $K$  of discriminant  $d_K < -4$ . We fix an algebraic closure of  $K$ . It follows from class field theory that there is a unique field  $K_{\mathcal{O}}$  with the property that the Artin map induces an isomorphism

$$\text{Gal}(K_{\mathcal{O}}/K) \xrightarrow{\sim} \text{cl}(\mathcal{O})$$

between the Galois group of  $K_{\mathcal{O}}/K$  and the ideal class group of  $\mathcal{O}$ . The field  $K_{\mathcal{O}}$  is called the *ring class field* for the order  $\mathcal{O}$ . If  $\mathcal{O}$  is the maximal order of  $K$ , then  $K_{\mathcal{O}}$  is the Hilbert class field of  $K$ , the maximal totally unramified abelian extension of  $K$ . In general, primes dividing  $[\mathcal{O}_K : \mathcal{O}]$  ramify in the ring class field.

The first main theorem of complex multiplication [21, Thm. 11.1] states that

$$K_{\mathcal{O}} = K(j(E)),$$

for any complex elliptic curve  $E$  with endomorphism ring  $\mathcal{O}$ . Furthermore, the minimal polynomial  $H_{\mathcal{O}}$  of  $j(E)$  over  $K$  actually has coefficients in  $\mathbf{Z}$ , and its degree is  $h(\mathcal{O}) = |\text{cl}(\mathcal{O})|$ . The polynomial  $H_{\mathcal{O}}$  is known as the *Hilbert class polynomial*. If  $p$  is a prime that splits completely in the extension  $K_{\mathcal{O}}/\mathbf{Q}$ , then  $H_{\mathcal{O}}$  splits into distinct linear factors in  $\mathbf{F}_p[X]$ . Its roots are the  $j$ -invariants of the elliptic curves  $E/\mathbf{F}_p$  with  $\text{End}(E) \cong \mathcal{O}$ , a set we denote  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$ . Let  $D = \text{disc}(\mathcal{O})$ . The primes that split completely in  $K_{\mathcal{O}}$  are precisely the primes  $p \nmid D$  that are the norm

$$p = N_{K/\mathbf{Q}} \left( \frac{t + v\sqrt{D}}{2} \right) = \frac{t^2 - v^2D}{4}$$

of an element of  $\mathcal{O}$ . The equation  $4p = t^2 - v^2D$  is often called the *norm equation*.

For a positive integer  $N$ , there is a unique extension  $K_{N,\mathcal{O}}$  of the ring class field  $K_{\mathcal{O}}$  such that the Artin map induces an isomorphism

$$\text{Gal}(K_{N,\mathcal{O}}, K_{\mathcal{O}}) \xrightarrow{\sim} (\mathcal{O}/N\mathcal{O})^*/\{\pm 1\}.$$

The field  $K_{N,\mathcal{O}}$  is the *ray class field of conductor  $N$  for  $\mathcal{O}$* . When  $\mathcal{O} = \mathcal{O}_K$ , this is simply the ray class field of conductor  $N$ , and for  $N = 1$  we recover the ring class field  $K_{\mathcal{O}} = K_{1,\mathcal{O}}$ . The ring class field  $K_R$  of the order  $R = \mathbf{Z} + N\mathcal{O}$  is a subfield of the ray class field  $K_{N,\mathcal{O}}$ . The Galois group of  $K_R/K_{\mathcal{O}}$  is isomorphic to

$$(\mathcal{O}/N\mathcal{O})^*/(\mathbf{Z}/N\mathbf{Z})^*,$$

the kernel of the map  $\varphi$  in (3).

The second main theorem of complex multiplication [21, Thm. 11.39] states that

$$K_{N,\mathcal{O}} = K_{\mathcal{O}}(x(E[N])),$$

where  $x(E[N])$  denotes the set of  $x$ -coordinates of the  $N$ -torsion points of an elliptic curve  $E$  with endomorphism ring  $\mathcal{O}$ . The Galois invariance of the *Weil pairing*  $E[N] \times E[N] \rightarrow \mu_N$  implies that the cyclotomic field  $\mathbf{Q}(\zeta_N)$  is contained in the ray class field  $K_{N,\mathcal{O}}$  (a fact that also follows directly from class field theory). In particular, a prime  $p$  that splits completely in  $K_{N,\mathcal{O}}$  also splits completely in  $\mathbf{Q}(\zeta_N)$ , and is therefore congruent to 1 modulo  $N$ .

**4.2. Primes that split completely in the ray class field.** We are specifically interested in primes  $p$  that split completely in the ray class field  $K_{l,\mathcal{O}}$ , where  $l$  is an odd prime. For such  $p$  we can achieve the desired setting for Algorithm 2.1, as depicted in Figure 1.

**Theorem 4.1.** *Let  $l > 2$  be prime, and let  $\mathcal{O} \not\subseteq \mathbf{Z}[i], \mathbf{Z}[\zeta_3]$  be an imaginary quadratic order that is maximal at  $l$ . Let  $R = \mathbf{Z} + l\mathcal{O}$  be the order of index  $l$  in  $\mathcal{O}$ . Let  $p$  be a prime that splits completely in the ray class field  $K_{l,\mathcal{O}}$ , but does not split completely in the ring class field for the order of index  $l^2$  in  $\mathcal{O}$ .*

- (1) *There are exactly  $h(\mathcal{O})$  different  $\mathbf{F}_p$ -isomorphism classes of elliptic curves  $E/\mathbf{F}_p$  with endomorphism ring  $\mathcal{O}$  that have  $E[l] \subset E(\mathbf{F}_p)$ .*
- (2) *There are exactly  $h(R)$  different  $\mathbf{F}_p$ -isomorphism classes of elliptic curves  $E/\mathbf{F}_p$  with endomorphism ring  $R$  that have an  $\mathbf{F}_p$ -rational  $l^2$ -torsion point.*

*Proof.* The inclusions  $K_{\mathcal{O}} \subseteq K_R \subseteq K_{l,\mathcal{O}}$  imply that both  $H_{\mathcal{O}}$  and  $H_R$  split into linear factors in  $\mathbf{F}_p$ . Each  $j$ -invariant in  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$ , resp.  $\text{Ell}_R(\mathbf{F}_p)$ , corresponds to two distinct isomorphism classes over  $\mathbf{F}_p$ , since these curves are ordinary and  $\mathcal{O} \not\subseteq \mathbf{Z}[i], \mathbf{Z}[\zeta_3]$ . We will show that exactly one of these satisfies (1), resp. (2).

Since  $p$  splits completely in the ray class field  $K_{l,\mathcal{O}}$ , we can factor  $p = \pi_p \bar{\pi}_p \in \mathcal{O}$  with  $\pi_p \equiv 1 \pmod{l\mathcal{O}}$ . Let  $E/\mathbf{F}_p$  be an elliptic curve with endomorphism ring  $\mathcal{O}$  whose Frobenius endomorphism corresponds to  $\pi_p$  under one of the two isomorphisms  $\text{End}(E) \xrightarrow{\sim} \mathcal{O}$ . Since  $\pi_p \equiv 1 \pmod{l\mathcal{O}}$ , we have  $E[l] \subset E(\mathbf{F}_p)$ . The Frobenius endomorphism of the non-isomorphic quadratic twist  $\tilde{E}/\mathbf{F}_p$  corresponds to  $-\pi_p$ , and we then have  $\#\tilde{E}(\mathbf{F}_p) = p + 1 - \text{tr}(-\pi_p) \equiv 2 \pmod{l}$ . For  $l \neq 2$  this implies that  $\tilde{E}$  has trivial  $l$ -torsion over  $\mathbf{F}_p$ , proving (1).

To prove (2), let  $E'/\mathbf{F}_p$  be a curve with endomorphism ring  $R$  that is  $l$ -isogenous to  $E$ . The Frobenius endomorphism of  $E'$  also corresponds to  $\pi_p$  under an isomorphism  $\text{End}(E') \xrightarrow{\sim} R$ . The cardinality of  $E'(\mathbf{F}_p)$  is thus equal to the cardinality of  $E(\mathbf{F}_p)$  and therefore divisible by  $l^2$ . However, since  $p$  does *not* split completely in the ring class field of index  $l^2$  in  $\mathcal{O}$ , we cannot have  $\pi_p \equiv 1 \pmod{lR}$ . It follows that  $E'[l] \not\subset E'(\mathbf{F}_p)$  and  $E'(\mathbf{F}_p)$  must contain a point of order  $l^2$ . As above, the quadratic twist of  $E'$  must have trivial  $l$ -torsion over  $\mathbf{F}_p$ , proving (2).  $\square$

Provided the order  $\mathcal{O}$  in Theorem 4.1 also satisfies  $h(\mathcal{O}) \geq l + 2$ , we can achieve the desired setting for Algorithm 2.1. We say such an order is *suitable* for  $l$ .

To determine the coefficients  $\Phi_l$  via the Chinese Remainder Theorem, we need to compute  $\Phi_l \bmod p$  for many primes  $p$  satisfying Theorem 4.1. We necessarily have  $p > l$ , since  $p \equiv 1 \pmod{l}$ , and the height bound [20] on the coefficients of  $\Phi_l$  implies that  $6l + O(1)$  primes suffice. We now show these primes exist and bound their size, assuming the GRH. For this purpose we define a *suitable family of orders*.

**Definition 4.2.** *Let  $S$  be the set of odd primes and let  $T$  be the set of all imaginary quadratic orders. A suitable family of orders is a function  $\mathcal{F} : S \rightarrow T$  such that:*

- (1) *for all  $l \in S$  the order  $\mathcal{F}(l)$  is suitable for  $l$ .*
- (2) *there exist effective constants  $c_1, c_2 \in \mathbf{R}_{>0}$  such that for all  $l \in S$  the bounds  $l + 2 \leq h(\mathcal{F}(l)) \leq c_1 l$  and  $l^2 \leq |\text{disc}(\mathcal{F}(l))| \leq c_2 l^2$  hold.*

**Example 4.3.** *Let  $\mathcal{F}(3) = \mathbf{Z}[\sqrt{-47}]$ , and for  $l > 3$  let  $\mathcal{F}(l)$  be the order  $\mathcal{O}$  of discriminant  $-7 \cdot 3^{2n}$ , where  $n$  is the least integer for which  $h(\mathcal{O}) = 2 \cdot 3^{n-1} \geq l + 2$ . Letting  $c_1 = 4$  and  $c_2 = 205$ , we see that  $\mathcal{F}$  is a suitable family of orders.*

**Theorem 4.4.** *Let  $\mathcal{F}$  be a suitable family of orders and let  $c_0 \in \mathbf{R}_{>0}$  be an arbitrary constant. Then for each prime  $l > 2$  the set of primes  $p$  for which  $l, \mathcal{O} = \mathcal{F}(l)$ , and  $p$  satisfy the conditions of Theorem 4.1 has positive density.*

*Assuming the GRH, there is an effective constant  $c \in \mathbf{R}_{>0}$  such that at least  $c_0 l^3 (\log l)^3$  of these primes are bounded by  $B = cl^6 (\log l)^4$ , for all primes  $l > 7$ .*

*Proof.* For a prime  $l > 2$ , let  $\mathcal{O} = \mathcal{F}(l)$  have fraction field  $K$ , and let  $u = [\mathcal{O}_K : \mathcal{O}]$ . The ray class field  $K_{l,\mathcal{O}}$  and the ring class field  $K_S$  for the order  $S = \mathbf{Z} + l^2\mathcal{O}$  are both invariant under the action of complex conjugation, hence both are Galois extensions of  $\mathbf{Q}$ . One finds that

$$\# \text{Gal}(K_{l,\mathcal{O}}/\mathbf{Q}) = 2(l-1)\left(l - \left(\frac{d_K}{l}\right)\right)h(\mathcal{O}) < 2l\left(l - \left(\frac{d_K}{l}\right)\right)h(\mathcal{O}) = \# \text{Gal}(K_S/\mathbf{Q}),$$

and the Chebotarëv density theorem [49, Thm. 13.4] yields the unconditional claim.

To prove the conditional claim, we apply an effective Chebotarëv bound to the extension  $K_{l,\mathcal{O}}/\mathbf{Q}$ , assuming the GRH for the Dedekind zeta function of  $K_{l,\mathcal{O}}$ .

The extension  $K_{l,\mathcal{O}}/K$  is abelian of conductor dividing  $lu$ , with degree  $nh(\mathcal{O})$ , where  $n \leq 2\#(\mathcal{O}/l\mathcal{O})^* \leq 2l^2$ . The  $\mathcal{O}_K$ -ideal  $\text{disc}(K_{l,\mathcal{O}}/\mathcal{O})$  is a divisor of  $(lu)^{nh(\mathcal{O})}$ , by Hasse's *Führerdiskriminantenproduktformel* [49, Thm. VII.11.9]. We then have

$$\begin{aligned} |\text{disc}(K_{l,\mathcal{O}}/\mathbf{Q})| &= |N_{K/\mathbf{Q}}(\text{disc}(K_{l,\mathcal{O}}/K)) \cdot \text{disc}(K/\mathbf{Q})^{[K_{l,\mathcal{O}}:K]}| \\ &\leq (lf)^{2nh(\mathcal{O})} |\text{disc}(K/\mathbf{Q})|^{nh(\mathcal{O})} \leq (c_2 l^4)^{nh(\mathcal{O})}, \end{aligned}$$

where  $\text{disc}(\mathcal{O}) \leq c_2 l^2$ . Using the bound  $h(\mathcal{O}) \leq c_1 l$ , Theorem 1.1 of [43] then yields

$$(7) \quad \left| \pi(x, K_{l,\mathcal{O}}/\mathbf{Q}) - \frac{\text{Li}(x)}{2nh(\mathcal{O})} \right| \leq c_3 \left( x^{1/2} \log(lx) + l^3 \log l \right),$$

where  $\pi(x, K_{l,\mathcal{O}}/\mathbf{Q})$  counts the primes up to  $x \in \mathbf{R}_{>0}$  that split completely in  $K_{l,\mathcal{O}}$ , and  $c_3 \in \mathbf{R}_{>0}$  is an effectively computable constant, independent of  $l$ .

If we now suppose  $x = cl^6 (\log l)^4$ , and apply  $\text{Li}(x) \sim x \log x$  and  $nh(\mathcal{O}) \leq c_1 l^3$ , we may choose  $c \in \mathbf{R}_{>0}$  so that  $\text{Li}(x)/(2nh(\mathcal{O}))$  is greater than the RHS of (7) by an arbitrarily large constant factor. In particular, for any  $c_4 \in \mathbf{R}_{>0}$  there is an effectively computable choice of  $c$  that ensures  $\pi(x, K_{l,\mathcal{O}}/\mathbf{Q}) \geq c_4 l^3 (\log l)^3$ , independent of  $l$ . Moreover, for the least such  $c$  we have  $c/c_4 \rightarrow 1$  as  $c_4 \rightarrow \infty$ .



We now show that most of these primes do not split completely in  $K_S$ . Any prime  $p$  that splits completely in  $K_{l,\mathcal{O}}$  must split completely in the ring class field for  $R = \mathbf{Z} + l\mathcal{O}$ . Putting  $D = \text{disc}(\mathcal{O})$ , we then have

$$(8) \quad 4p = t^2 - v^2 l^2 D,$$

with  $t, v \in \mathbf{Z}_{>0}$  and  $t \equiv 2 \pmod{l}$ . If  $v \not\equiv 0 \pmod{l}$ , then  $p$  cannot split completely in  $K_S$ . For  $p \leq cl^6(\log l)^4$ , we have  $v \leq 2c^{1/2}l(\log l)^2$  and  $t \leq 2c^{1/2}l^3(\log l)^2$ , since  $D \geq l^2$ , hence there are at most  $2c^{1/2}(\log l)^2$  positive  $v \equiv 0 \pmod{l}$ , and at most  $2c^{1/2}l^2(\log l)^2 + 1$  positive  $t \equiv 2 \pmod{l}$ , that satisfy (8).

It follows that no more than  $4cl^2(\log l)^4 + 2c^{1/2}(\log l)^2$  primes  $p \leq cl^6(\log l)^4$  split completely in  $K_S$ . For a sufficiently large choice of  $c_4$ , we can choose  $c$  so that  $\pi(x, K_{l,\mathcal{O}}/\mathbf{Q}) = \pi(cl^6(\log l)^4, K_{l,\mathcal{O}}/\mathbf{Q}) \geq c_4 l^3(\log l)^3$  and also

$$c_4 l^3(\log l)^3 - 4cl^2(\log l)^4 - 2c^{1/2}(\log l)^2 > c_0 l^3(\log l)^3,$$

provided  $l > 7$ , since  $l/(\log l)$  is bounded above 4 for primes  $l > 7$ .  $\square$

Theorem 4.4 guarantees we can obtain a sufficient number of primes  $p$  for use with Algorithm 2.1. In fact, as is typical for such bounds, it provides far more than we need. The task of finding these primes is addressed in Section 6.1.

**4.3. Computing the CM action.** The Galois action of  $\text{Gal}(K_{\mathcal{O}}/K) \cong \text{cl}(\mathcal{O})$  on the set  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$  may be explicitly computed using isogenies, as described in [2]. Let the prime  $p$  split completely in the ring class field  $K_{\mathcal{O}}$ , and let  $E/\mathbf{F}_p$  be an elliptic curve with  $\text{End}(E) \cong \mathcal{O}$ . Fixing an isomorphism  $\text{End}(E) \xrightarrow{\sim} \mathcal{O}$ , for each invertible  $\mathcal{O}$ -ideal  $\mathfrak{a}$  we define

$$E[\mathfrak{a}] = \{P \in E(\overline{\mathbf{F}}_p) \mid \forall \tau \in \mathfrak{a} : \tau(P) = 0\},$$

the ‘ $\mathfrak{a}$ -torsion’ subgroup of  $E$ . The subgroup  $E[\mathfrak{a}]$  is the kernel of a separable isogeny  $E \rightarrow E/E[\mathfrak{a}]$  of degree  $[\mathcal{O} : \mathfrak{a}]$ , with  $\text{End}(E/E[\mathfrak{a}]) \cong \mathcal{O}$ . This yields a group action

$$j(E)^{\mathfrak{a}} = j(E/E[\mathfrak{a}]),$$

in which the ideal group of  $\mathcal{O}$  acts on the set  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$ . This action factors through the class group, and the  $\text{cl}(\mathcal{O})$ -action is transitive and free. Equivalently,  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$  is a *torsor* for  $\text{cl}(\mathcal{O})$ ; for each pair  $(j_1, j_2)$  of elements in  $\text{Ell}_{\mathcal{O}}$  there is a unique element of  $\text{cl}(\mathcal{O})$  whose action sends  $j_1$  to  $j_2$ .

Now let  $\mathfrak{l}_0$  be an invertible  $\mathcal{O}$ -ideal of prime norm  $l_0 \neq p$ . The curves  $E$  and  $E/E[\mathfrak{l}_0]$  are  $l_0$ -isogenous, hence

$$\Phi_{l_0}(j_0, j_0^{l_0}) = 0,$$

where  $j_0 = j(E)$ . To compute the action of  $\mathfrak{l}_0$ , we need to find the corresponding root of  $\Phi_{l_0}(X, j_0) \in \mathbf{F}_p[X]$ . We assume that  $\Phi_{l_0}(X, Y)$  is known, either via one of the algorithms from the introduction, or by a previous application of Algorithm 6.1. The polynomial  $\Phi_{l_0}(X, j_0) \in \mathbf{F}_p[X]$  has either 1 or 2 roots that lie in  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$ , depending on whether  $l_0$  ramifies or splits (it is not inert). These roots correspond to the actions of  $\mathfrak{l}_0$  and its inverse  $\mathfrak{l}_0^{-1}$ , which coincide when  $l_0$  ramifies.

Our fixed isomorphism  $\text{End}(E) \xrightarrow{\sim} \mathcal{O}$  maps the Frobenius endomorphism of  $E$  to an element  $\pi_p \in \mathcal{O} \subset \mathcal{O}_K$  with norm  $p$ . We then have the norm equation

$$(9) \quad 4p = t^2 - v^2 d_K,$$

where  $t = \text{tr}(\pi_p)$ , and  $v$  is the index of  $\mathbf{Z}[\pi_p]$  in  $\mathcal{O}_K$ . When  $l_0$  does not divide  $v$ , the order  $\mathbf{Z}[\pi_p]$  is maximal at  $l_0$  and the only roots of  $\Phi_{l_0}(X, j_0)$  over  $\mathbf{F}_p$  are those

in  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$ . Otherwise  $\Phi_{l_0}(X, j_0)$  has  $l + 1$  roots in  $\mathbf{F}_p$ , and those in  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$  lie on the surface of the  $l_0$ -volcano containing  $j$ , as described in [30]. The roots on the surface can be readily distinguished, as in [54, §4], for example, but typically we choose  $p$  with  $l_0 \nmid v$  so that every root of  $\Phi_{l_0}(X, j_0)$  in  $\mathbf{F}_p$  is on the surface.

When  $l_0$  splits and does not divide  $v$ , the actions of  $\mathfrak{l}_0$  and  $\mathfrak{l}_0^1$  may be distinguished as described in [11, §5] and [31, §3]. The kernels of the two  $l_0$ -isogenies are subgroups of  $E[l_0]$ . A standard component of the SEA algorithm computes a polynomial  $F_{l_0}(X)$ , whose roots are the abscissa of the points in one of these kernels [22, 51]. In our setting  $l_0$  splits in  $\mathbf{Z}[\pi_p]$ , and provided  $l_0 \nmid v$ , the action of  $\pi_p$  on  $E[l_0]$  has two distinct eigenvalues corresponding to the two kernels. Expressing the ideal  $\mathfrak{l}_0$  in the form  $(l_0, c + d\pi_p)$  yields the eigenvalue  $\lambda = -c/d \pmod{l_0}$ . We may then use  $F_{l_0}(X)$  to test whether  $\pi_p$ 's action is equivalent to multiplication by  $\lambda$  in the corresponding kernel. See [11] for an example and further details.

As a practical optimization (see Section 6.6), we avoid the need to ever make this distinction. The asymptotic complexity of computing the action of  $\mathfrak{l}_0$  is the same in any case.

**Lemma 4.5.** *Let  $l_0$  and  $p$  be distinct odd primes, and let  $\mathcal{O} \neq \mathbf{Z}[i], \mathbf{Z}[\zeta_3]$  be an imaginary quadratic order. Let  $j_0 = j(E) \in \text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$ , fix an isomorphism  $\text{End}(E) \xrightarrow{\sim} \mathcal{O}$ , and let  $\pi_p \in \mathcal{O}$  denote the image of the Frobenius endomorphism. Let  $\mathfrak{l}_0$  be an invertible  $\mathcal{O}$ -ideal of norm  $l_0$ , and assume  $\mathbf{Z}[\pi_p]$  is maximal at  $l_0$ .*

*Given  $\Phi_{l_0} \in \mathbf{F}_p[X, Y]$ , the  $j$ -invariant  $j_0^{l_0}$  may be computed using an expected  $O(l_0^2 + M(l_0) \log p)$  operations in  $\mathbf{F}_p$ .*

Here  $M(n)$  denotes the complexity of multiplying two polynomials of degree less than  $n$ , as in [56, Def. 8.26]. Naïvely,  $M(n) = O(n^2)$ , Karatsuba's algorithm yields  $M(n) = O(n^{\log_2 3})$ , and methods based on the fast Fourier transform (FFT) achieve  $M(n) = O(n \log n \log \log n)$ .

*Proof.* We first compute  $\gcd(X^p - X, \Phi_{l_0}(X, j_0))$ , the product of the distinct linear factors of  $\Phi_{l_0}(X, j_0)$  over  $\mathbf{F}_p$ . Instantiating  $f(X) = \Phi_{l_0}(X, j_0)$  uses  $O(l_0^2)$  operations in  $\mathbf{F}_p$ , exponentiating  $X^p \pmod f$  uses  $O(M(l_0) \log p)$  operations in  $\mathbf{F}_p$ , and the fast Euclidean algorithm [56, §11.1] obtains  $\gcd(X^p - X, f)$  using  $O(M(l_0) \log l_0) = O(l_0^2)$  operations in  $\mathbf{F}_p$ . This gcd has degree at most 2, since  $\mathbf{Z}[\pi_p]$  is maximal to  $l_0$ , and we may find its roots using an expected  $O(\log p)$   $\mathbf{F}_p$ -operations [56, Cor. 14.16].

The desired root  $j_0^{l_0}$  is then distinguished as outlined above. We first compute the eigenvalue  $\lambda = c/d \pmod{l_0}$ , where  $\mathfrak{l}_0 = (l_0, c + d\pi_p)$ , using  $O(l_0^2)$  bit operations. Applying [9, Thm. 2.1], the kernel polynomial  $F_{l_0}(X)$  can be computed using  $O(M(l_0))$  operations in  $\mathbf{F}_p$ . To compare  $(X^p, Y^p)$  to the scalar multiple  $\lambda \cdot (X, Y)$ , we compute  $X^p, Y^p$ , and the required division polynomials  $\psi_n(X, Y)$ , modulo  $F_{l_0}(X)$  and the curve equation for  $E$ , as in the SEA algorithm [7, Ch. VII]. This uses  $O((\log l_0 + \log p)M(l_0)) = O(l_0^2 + M(l_0) \log p)$  operations in  $\mathbf{F}_p$ .  $\square$

## 5. MAPPING THE CM TORSOR

The previous section made explicit the Galois action corresponding to an element of  $\text{cl}(\mathcal{O}) \cong \text{Gal}(K_{\mathcal{O}}/K)$  represented by an ideal  $\mathfrak{l}_0$  of prime norm  $l_0$ . We now use this to enumerate the set  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$ , and at the same time compute a map that explicitly identifies the action of each element of  $\text{cl}(\mathcal{O})$ . To do this efficiently it is critical to work with generators whose norms are small, since the cost of computing the action of  $\mathfrak{l}_0$  increases quadratically with its norm.

**5.1. Polycyclic presentations.** As a finite abelian group, each element of  $\text{cl}(\mathcal{O})$  can be uniquely represented using a basis. However, as noted in [54, §5.3], the norms arising in a basis may need to be much larger than those in a set of generators. Thus we are led to consider polycyclic presentations.

Let  $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_k)$  be a sequence of generators for  $G$ , and let  $G_i = \langle \alpha_1, \dots, \alpha_i \rangle$  denote the subgroup generated by  $\alpha_1, \dots, \alpha_i$ . The composition series

$$1 = G_0 \leq G_1 \leq \dots \leq G_{k-1} \leq G_k = G,$$

is then polycyclic, meaning that each quotient  $G_{i+1}/G_i$  is a cyclic group. The sequence  $r(\boldsymbol{\alpha}) = (r_1, \dots, r_k)$  of *relative orders* for  $\boldsymbol{\alpha}$  is defined by

$$r_i = |G_i : G_{i-1}|.$$

Each  $r_i$  necessarily divides  $|\alpha_i|$ , and for  $i > 1$  we typically have  $r_i < |\alpha_i|$ . The sequences  $\boldsymbol{\alpha}$  and  $r(\boldsymbol{\alpha})$  allow us to uniquely represent each  $\beta \in G$  in the form

$$(10) \quad \beta = \boldsymbol{\alpha}^{\boldsymbol{x}} = \alpha_1^{x_1} \cdots \alpha_k^{x_k},$$

where  $\boldsymbol{x} = (x_1, \dots, x_k)$  with  $0 \leq x_i < r_i$ . The vector  $s(\boldsymbol{\alpha}, i) = \boldsymbol{x}$  for which  $\alpha_i^{r_i} = \boldsymbol{\alpha}^{\boldsymbol{x}}$  has  $x_j = 0$  for  $j \geq i$ , and is called a *power relation*, see [38, §8.1]. A generic algorithm to compute  $r(\boldsymbol{\alpha})$  and the  $s(\boldsymbol{\alpha}, i)$  can be found in [54, Alg. 2.1].

The vector  $\boldsymbol{x}$  in (10) is the discrete logarithm or *exponent vector* of  $\beta$ . We let

$$X(\boldsymbol{\alpha}) = \{\boldsymbol{x} \in \mathbf{Z}^k : 0 \leq x_i < r_i\},$$

and note that the map  $\boldsymbol{x} \mapsto \boldsymbol{\alpha}^{\boldsymbol{x}}$  defines a bijection from  $X(\boldsymbol{\alpha})$  to  $G$ .

We now consider the case  $G = \text{cl}(\mathcal{O})$ , where  $\mathcal{O}$  is an order in a quadratic field  $K$  of discriminant  $d_K < -4$ . Let  $\mathcal{P} = (p_1, p_2, p_3, \dots)$  be an increasing sequence of primes with the property that  $\text{cl}(\mathcal{O})$  is generated by the classes of invertible ideals with norms in  $\mathcal{P}$ . By Dirichlet's density theorem [21, Thm. 9.12], any sequence containing all but a finite set of primes works, and from [21, Cor. 7.17] we know that some finite prefix of  $\mathcal{P}$  actually suffices. For  $\mathcal{O} = \mathcal{O}_K$  we may take  $\mathcal{P}$  to be the sequence of primes less than  $|d_K/3|^{1/2}$ , by [14, Prop. 9.5.2].

There is a unique lexicographically minimal subsequence  $(l_1, \dots, l_k)$  of  $\mathcal{P}$  that corresponds to a polycyclic sequence  $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_k)$  for  $\text{cl}(\mathcal{O})$  in which  $\alpha_i$  is represented by an ideal of norm  $l_i$  and  $r(\boldsymbol{\alpha})$  has  $r_i > 1$ . When  $l_i$  splits there are two possibilities for  $\alpha_i$ . To fix a choice, let  $\alpha_i$  be the ideal class represented by the unique binary quadratic form  $ax^2 + bxy + cy^2$  of discriminant  $D = \text{disc}(\mathcal{O})$  with  $a = l_i$  and  $b$  nonnegative [14, §3.4], corresponding to the ideal  $\mathfrak{f}_i = (l_i, (-b + \sqrt{D})/2)$ .

We call  $\boldsymbol{\alpha}$  the *polycyclic presentation* of  $\text{cl}(\mathcal{O})$  determined by  $\mathcal{P}$ . We use  $l(\boldsymbol{\alpha})$  to denote the sequence of norms  $(l_1, \dots, l_k)$ , but note that this also depends on  $\mathcal{P}$ ; each  $l_i$  is the least prime in  $\mathcal{P}$  that is the norm of an ideal in  $\alpha_i$ .

We may compute  $\boldsymbol{\alpha}$  by applying [54, Alg. 2.1] to an implicit sequence of generators  $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \gamma_3, \dots)$  corresponding to the subsequence of  $\mathcal{P}$  for which there exists an invertible  $\mathcal{O}$ -ideal of norm  $p_i$ . The algorithm computes  $r_i$  for each  $\gamma_i$  in turn, and if we find that  $r_i > 1$ , we append  $\gamma_i$  to an initially empty vector  $\boldsymbol{\alpha}$ . We terminate when  $\prod r_i = h(\mathcal{O})$ , a value which we assume has been precomputed.

The computation of  $\boldsymbol{\alpha}$  uses  $|G| = h(\mathcal{O})$  group operations in  $G = \text{cl}(\mathcal{O})$ , and creates a table  $T : X(\boldsymbol{\alpha}) \rightarrow G$  that stores  $|G| = h(\mathcal{O})$  group elements [54, Prop. 6]. Using binary quadratic forms to represent  $\text{cl}(\mathcal{O})$ , the group operation has bit-complexity  $O(\log^2 |\text{disc}(\mathcal{O})|)$ , as shown in [5], and each element may be stored in  $O(\log |\text{disc}(\mathcal{O})|)$  space. Evaluating  $T(\boldsymbol{x})$ , or  $T^{-1}(\beta)$ , has bit-complexity  $O(\log |G|)$ .

**5.2. Suitable presentations.** When  $\mathcal{P}$  is the sequence of all primes, the norms  $l(\boldsymbol{\alpha})$  for the polycyclic presentation  $\boldsymbol{\alpha}$  of  $\text{cl}(\mathcal{O})$  determined by  $\mathcal{P}$  are as small as possible. However, when working in the finite field  $\mathbf{F}_p$  we may wish to ensure that each norm  $l_i$  does not divide  $v = [\mathcal{O}_K : \mathbf{Z}[\pi_p]]$ , as noted in Section 4.3. This is achieved by excluding from  $\mathcal{P}$  primes that divide  $v$ , and we call the corresponding  $\boldsymbol{\alpha}$  the presentation of  $\text{cl}(\mathcal{O})$  *suitable* for  $p$ . This may cause us to use norms that are slightly larger than optimal. We now show that, provided we work with a family of orders that satisfies certain (easily met) constraints, the norms in every suitable presentation are quite small, assuming the GRH.

**Theorem 5.1.** *Let  $c_3 \in \mathbf{R}_{>0}$  be a fixed constant, and let  $\mathcal{F}$  be a suitable family of orders with the following additional property: if  $\mathcal{O}$  is an order in  $\mathcal{F}$  whose fraction field  $K$  has discriminant  $d_K$ , and  $s_{\mathcal{O}}$  denotes the square-free part of  $[\mathcal{O}_K : \mathcal{O}]$ , then  $s_{\mathcal{O}}$  is coprime to  $2d_K$  and both  $s_{\mathcal{O}}$  and  $|d_K|$  are bounded by  $c_3$ .*

*Then under the GRH, for every  $\mathcal{O}$  in  $\mathcal{F}$  and every prime  $p$  that splits completely in  $K_{\mathcal{O}}$ , the presentation  $\boldsymbol{\alpha}$  of  $\text{cl}(\mathcal{O})$  suitable for  $p$  has norms  $l(\boldsymbol{\alpha})$  for which*

$$\max l(\boldsymbol{\alpha}) \leq c\omega(v) \log(\omega(v) + 1),$$

*where  $v$  is defined by  $4p = t^2 - v^2 \text{disc}(\mathcal{O})$ , the function  $\omega(v)$  counts the distinct prime factors of  $v$ , and  $c$  is an effective constant that depends only on  $c_3$ .*

*Proof.* Let  $\mathcal{O}$ ,  $p$ ,  $v$ , and  $\boldsymbol{\alpha}$  be as above. Let  $R$  be the order of index  $s_{\mathcal{O}}^2$  in  $\mathcal{O}_K$ , and let  $\boldsymbol{\gamma}$  be the presentation of  $R$  determined by the increasing sequence of primes that do not divide  $v$ . It follows from Theorem 3.3 that  $\max l(\boldsymbol{\alpha}) \leq \max l(\boldsymbol{\gamma})$ .

Let  $K_R$  be the ring class field for  $R$ , and let  $\pi_C(x, K_R/\mathbf{Q})$  count the primes bounded by  $x \in \mathbf{R}_{>0}$  whose Frobenius symbol (under the Artin map) lies in the conjugacy class  $C$  of  $\text{Gal}(K_R/\mathbf{Q})$ . We may bound  $[K_R:\mathbf{Q}]$ ,  $\#\text{Gal}(K_R/\mathbf{Q})$ , and  $\text{disc}(K_R)$  by constants that depend only on  $c_3$ , independent of  $\mathcal{O}$ . Under the GRH, the Chebotarëv bound of [43, Thm. 1.1] then yields

$$\pi_C(x, K_R/\mathbf{Q}) \geq c_4 x / \log x,$$

for some effective constant  $c_4 \in \mathbf{R}_{>0}$  and all  $x > 2$ , where  $c_4$  depends only on  $c_3$ . For an effective constant  $c$  depending on  $c_3$ , setting  $x = c\omega(v) \log(\omega(v) + 1)$  yields  $\pi_C(x, K_R/\mathbf{Q}) > \omega(v)$ . In this case the Frobenius symbol of at least one prime not dividing  $v$  lies in  $C$ , and this applies to every  $C$ . It follows that every class in  $\text{cl}(R)$  contains an element whose norm is a prime bounded by  $x$  that does not divide  $v$ . We then have  $\max l(\boldsymbol{\alpha}) \leq \max l(\boldsymbol{\gamma}) \leq x = c\omega(v) \log(\omega(v) + 1)$ , as desired.  $\square$

The family of orders in Example 4.3 satisfies the requirements of Theorem 5.1. We note that provided  $\log p = O(\log l)$ , we have  $\omega(v) = O(\log l / \log \log l)$ , and the theorem then yields an  $O(\log l)$  bound on the norms  $l(\boldsymbol{\alpha})$ . This is sharper than the more general  $O(\log^2 l)$  bound implied by [1]. In fact, by [34, Thm. 431], one expects  $\omega(v) = O(\log \log p)$ , which yields a bound of  $O(\log \log l \log \log l)$ .

**5.3. Realizing the CM torsor.** We now consider how to explicitly map  $\text{cl}(\mathcal{O})$  to the torsor  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$ , so that we may then compute the action of any element or subgroup of  $\text{cl}(\mathcal{O})$  on any element of  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$ , *without needing to compute any further isogenies*. We use the presentation  $\boldsymbol{\alpha}$  of  $\text{cl}(\mathcal{O})$  suitable for  $p$ , and the table  $T : X(\boldsymbol{\alpha}) \rightarrow \text{cl}(\mathcal{O})$  described in Section 5.1. As above, we have  $\boldsymbol{\alpha} = ([l_1], \dots, [l_k])$ , with norms  $l(\boldsymbol{\alpha}) = (l_1, \dots, l_k)$  and relative orders  $r(\boldsymbol{\alpha}) = (r_1, \dots, r_k)$ . We assume the modular polynomials  $\Phi_{l_1}, \dots, \Phi_{l_k}$  are known, since the  $l_i$  are small.

To enumerate  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$  we use [54, Alg. 1.3], but we augment this algorithm to also compute an explicit bijection  $\phi: \text{cl}(\mathcal{O}) \rightarrow \text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$  in which  $[\mathbf{a}] \in \text{cl}(\mathcal{O})$  corresponds to  $j_0^{\mathbf{a}}$ . Given  $j_0 \in \text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$ , we compute a path of  $l_k$ -isogenies

$$(11) \quad j_0 \xrightarrow{l_k} j_1 \xrightarrow{l_k} j_2 \xrightarrow{l_k} \cdots \xrightarrow{l_k} j_{r_k-1},$$

where the  $j_i$  are distinct elements of  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$ . As explained in Section 4.3, each step in this path is computed by finding a root  $j_i$  of  $\Phi(X, j_{i-1})$  that lies in  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$ . When  $l_k$  splits in  $K$  we have two choices for  $j_1$ , and the correct choice may be determined using a kernel polynomial as outlined in Section 4.3. For  $i > 1$  we use the polynomial  $\Phi(X, j_{i-1})/(X - j_{i-2})$ , which has exactly one root  $j_i \in \text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$ .

When  $k > 1$ , the enumeration of  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$  proceeds recursively: for each  $j_i$  in (11) we compute a path of  $l_{k-1}$  isogenies containing  $r_{k-1}$  distinct  $j$ -invariants. Eventually, every element of  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$  is enumerated exactly once [54, Prop. 5]. For each  $j_n \in \text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$  we also compute a vector  $\mathbf{x} \in X(\boldsymbol{\alpha})$  that describes the path used to reach  $j_n$  from  $j_0$ , where  $x_i$  indicates the number of steps taken on an  $l_i$ -isogeny path. By correctly choosing the direction of each path, we ensure that each  $j_n$  is the image of  $j_0$  under the  $\text{cl}(\mathcal{O})$ -action of  $\boldsymbol{\alpha}^{\mathbf{x}} = \alpha_1^{x_1} \cdots \alpha_k^{x_k}$ . This yields the desired bijection  $\phi$ ; since  $\boldsymbol{\alpha}^{\mathbf{x}}$  uniquely represents some  $\beta \in \text{cl}(\mathcal{O})$ , we may set  $\phi(\boldsymbol{\alpha}^{\mathbf{x}}) = j_n$ , a process facilitated by the map  $T: X(\boldsymbol{\alpha}) \rightarrow \text{cl}(\mathcal{O})$ .

The bijection  $\phi$  allows us translate any computation in the group  $\text{cl}(\mathcal{O})$  to the torsor  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$ . In particular, by enumerating the cyclic subgroup  $H \subseteq \text{cl}(\mathcal{O})$  generated by  $[\mathfrak{l}]$ , where  $\mathfrak{l}$  is an ideal of norm  $l$ , we obtain the  $l$ -isogeny cycle containing  $j_0$ , corresponding to the surface of one of the  $l$ -volcanoes in Figure 1. Doing the same for each coset of  $H$  partitions  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$  into  $l$ -isogeny cycles.

This may also be applied to the order  $R = \mathbf{Z} + l\mathcal{O}$ . After obtaining a bijection from  $\text{cl}(R)$  to  $\text{Ell}_R(\mathbf{F}_p)$ , we enumerate the kernel of the map  $\varphi: \text{cl}(R) \rightarrow \text{cl}(\mathcal{O})$  from the exact sequence of (3). Here we use one of the generators of norm  $l^2$  guaranteed by Lemma 3.2. Enumerating the cosets of  $\ker \varphi$  then partitions  $\text{Ell}_R(\mathbf{F}_p)$  into  $l^2$ -isogeny cycles of siblings with a common  $l$ -isogenous parent in  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$ .

## 6. THE ALGORITHM

We now present our algorithm to compute the modular polynomial  $\Phi_l$  using the Chinese Remainder Theorem (CRT). Algorithm 6.1 follows the standard pattern of a CRT-based algorithm; the details lie in Algorithm 6.2, which selects a set of primes  $S$ , and in Algorithm 2.1, which computes  $\Phi_l$  modulo each prime  $p \in S$ .

The computation of  $\Phi_l \in \mathbf{Z}[X, Y]$  may be viewed as a special case of computing  $\Phi_l \in (\mathbf{Z}/m\mathbf{Z})[X, Y]$ , where  $m$  is the product of the primes in  $S$ . The choice of  $S$  ensures that this  $m$  is large enough to uniquely determine  $\Phi_l \in \mathbf{Z}[X, Y]$ .

**Algorithm 6.1.** Let  $l$  be an odd prime, let  $m$  be a positive integer, and let  $\mathcal{O} = \mathcal{F}(l)$  lie in a suitable family of orders  $\mathcal{F}$ . Compute  $\Phi_l \in (\mathbf{Z}/m\mathbf{Z})[X, Y]$  as follows:

1. Compute the Hilbert class polynomial  $H_{\mathcal{O}} \in \mathbf{Z}[X]$ .
2. Select a set of primes  $S$  with Algorithm 6.2, using  $l$  and  $\mathcal{O}$ .
3. Perform CRT precomputation using  $S$ .
4. For each prime  $p \in S$ :
  - a. Compute  $\Phi_l \bmod p$  with Algorithm 2.1, using  $\mathcal{O}$  and  $H_{\mathcal{O}}$ .
  - b. Update CRT data using  $\Phi_l \bmod p$ .
5. Perform CRT postcomputation.
6. Output  $\Phi_l \in (\mathbf{Z}/m\mathbf{Z})[X, Y]$ .

The suitable family of orders  $\mathcal{F}$  is as defined in Section 4.2, see Definition 4.2. The polynomial  $H_{\mathcal{O}}$  computed in Step 1 may be obtained using any of several algorithms whose running time is quasi-linear in  $\text{disc}(\mathcal{O})$ , including [11, 23, 54].

**6.1. Selecting primes.** The primes  $p$  in the set  $S$  selected by Algorithm 6.1 must satisfy the conditions of Theorem 4.1 in order to use them in Algorithm 2.1. We require  $p$  to split completely in the ray class field  $K_{l,\mathcal{O}}$ , but to not split completely in the ring class field for the order  $\mathbf{Z} + l^2\mathcal{O}$ . Equivalently, we need  $p \nmid D$  to satisfy

$$(12) \quad 4p = t^2 - v^2l^2D,$$

with  $t \equiv 2 \pmod{l}$  and  $l \nmid v$ , where  $D = \text{disc}(\mathcal{O})$ . To apply the CRT, we also require

$$(13) \quad \prod_{p \in S} \log p \geq 4|c|,$$

for every coefficient  $c$  of  $\Phi_l \in \mathbf{Z}[X, Y]$ . From [13], we use the explicit bound

$$(14) \quad B_l = 6l \log l + 18l$$

on the logarithmic height of  $\Phi_l$  to achieve this. We then have  $\#S = O(l)$ , by (12).

Heuristically, it is easy to find primes that satisfy (12). If  $D \equiv 1 \pmod{8}$ , fix  $v = 2$ , otherwise fix  $v = 1$ . Then, for increasing  $t \equiv 2 \pmod{l}$  with the correct parity, test whether  $p = (t^2 - v^2l^2D)/4$  is prime. We expect to need  $O(l \log l)$  primality tests, and each can be accomplished in time polynomial in  $\log l$ , although typically  $p$  is small enough to make an attempted factorization more efficient. We could obtain slightly smaller  $p$ 's by letting  $v$  vary, but it is more convenient to fix  $v$  so that we can use the same presentation of  $\text{cl}(\mathcal{O})$  and  $\text{cl}(R)$  for every  $p$ . This approach is easy to implement and very fast in practice.

However, in order to prove Theorem 1 we must take a more cautious approach. Even assuming the GRH, we cannot guarantee we will find *any* primes with a fixed value of  $v$ . On the other hand, Theorem 4.4 implies that if we construct random integers  $p \leq x$  satisfying (12), for sufficiently large  $x$  we have  $p$  prime with probability  $\Omega(1/\log x)$ , and under the GRH,  $x = O(l^6(\log l)^4)$  is large enough. Additionally, we would like to avoid  $v$ 's with many prime factors, so that we may more profitably apply Theorem 5.1. The restriction  $\omega(v) \leq 2 \log(\log v + 3)$  eliminates an asymptotically negligible proportion of the integers  $v \in [1, x]$  (see Lemma 8.1).

We now present Algorithm 6.2, emphasizing that its purpose is to facilitate the proof of Theorem 1. In practice we use the heuristic procedure described above.

**Algorithm 6.2.** Let  $l$  be an odd prime, let  $D < -4$  be a discriminant, and let  $B_l$  be as in (14). Construct the set  $S$  as follows:

1. Set  $n \leftarrow (B_l + 2 \log 2) / \log(l^2|D|/4)$  and then  $x \leftarrow 4l^2|D|n \log n$ .  
Set  $b \leftarrow 0$  and  $S \leftarrow \emptyset$ .
2. Set  $T \leftarrow 2x^{1/2}$  and  $V \leftarrow 2x^{1/2}l^{-1}|D|^{-1/2}$ .
3. Repeat  $\lceil 2N \log x \rceil$  times:
  - a. Construct an integer  $p = (t^2 - v^2l^2D)/4$  using uniformly random integers  $v \in [1, V]$  and  $t \in [1, T]$ , subject to  $l \nmid v$ ,  $t \equiv 2 \pmod{l}$ , and  $t \equiv vD \pmod{2}$ .
  - b. If  $\omega(v) > 2 \log(\log v + 3)$  then go to Step 3d.
  - c. If  $p \notin S$  and  $p$  is prime then set  $S \leftarrow S \cup \{p\}$  and  $b \leftarrow b + \log p$ .
  - d. If  $b > B_l + 2 \log 2$  then output  $S$  and terminate.
4. Set  $x \leftarrow 2x$  and go to Step 2.

In Step 3a, the integer  $t$  is generated as  $t = al + 2$  using a uniformly random integer  $a \in [0, V/l - 2]$ . The computation of  $\omega(v)$  in Step 3b is performed by factoring  $v$ . Note that  $\#S \leq n$ , and  $p \leq x$  for all  $p \in S$ .

**Lemma 6.3.** *Let  $\mathcal{F}$  be a suitable family of orders, let  $l$  be an odd prime and let  $D = \text{disc}(\mathcal{F}(l))$ . Given inputs  $l$  and  $D$ , the expected running time of Algorithm 6.2 is finite. Under the GRH we also have the following:*

- (1) *The expected running time is  $O(l^{1+\varepsilon})$ , for any  $\varepsilon \in \mathbf{R}_{>0}$ .*
- (2) *There is a constant  $c < 1$  such that for all  $l > 7$  and  $k \in \mathbf{Z}_{>0}$ , the algorithm terminates with  $\log x \leq (6 + k) \log l$  with probability at least  $1 - c^{-k \log l}$ .*

*Proof.* To analyze Algorithm 6.2, we count the number of times Step 4 is executed, referring to the period between each execution as an iteration. By Theorem 4.4, the set of primes that satisfy Theorem 4.1, equivalently, those that satisfy (12), has positive density. Here we may use the natural density, via [40, Thm. 4.3.e]. For every fixed odd prime  $l$ , this implies a lower bound of  $\Omega(1/\log x)$  on the probability that a random integer in  $[1, x]$  is a prime that satisfies (12). Each integer  $p$  tested by Algorithm 6.2 necessarily satisfies (12), hence such a  $p$  is prime with probability  $\Omega(1/\log x)$ . By Lemma 8.1 in the appendix, the probability that a candidate  $p$  is skipped due to the test in Step 3b is  $o(1)$ . This implies that for all sufficiently large  $x$ , the probability that Algorithm 6.2 terminates in a given iteration is bounded above zero, and the expected running time is finite.

Now assume the GRH and let  $l > 7$ . Applying Theorem 4.4 with  $c_0 = 1$ , there are at least  $l^3(\log l)^3$  primes  $p \leq c_1 l^6 (\log l)^4$  that satisfy (12), for some constant  $c_1 \in \mathbf{R}_{>0}$  that does not depend on  $l$ . Let  $x_0$  be the least value of  $x \geq c_1 l^6 (\log l)^4$ . When  $x = x_0$  we have  $VT/l \leq 8c_1 l^3 (\log l)^4$ , since  $|D| \geq l^2$ , and the probability that a given primality test succeeds is at least  $8c_1/\log l \geq c_2/\log x$ , for some constant  $c_2 \in \mathbf{R}_{>0}$ . From the inequality (7) in the proof of Theorem 4.4, one finds that this holds for all  $x \geq x_0$ , with the same constants. As above, Step 3b has negligible impact, and for  $x \geq x_0$  the probability that the algorithm terminates in a given iteration is at least  $c$ , for some constant  $c \in \mathbf{R}_{>0}$  independent of  $l$  and  $x$ .

We now consider the running time as a function of  $l$ , fixing an arbitrary  $\varepsilon \in \mathbf{R}_{>0}$ . It takes  $O(\log l)$  iterations to achieve  $x = x_0$ , assuming that we don't terminate earlier, and we execute Steps 3b and 3c a total of  $O(l(\log l)^2)$  times during this process. The computation of  $\omega(v)$  and the primality test of  $p$  can both be achieved in expected time subexponential in  $\log x$ , by [46], yielding an  $O(l^{1+\varepsilon})$  bound on the time to reach  $x = x_0$ , since  $\log x_0 = O(\log l)$ .

For  $x \geq x_0$ , the probability of reaching each subsequent iteration declines exponentially, while the cost of Steps 3b and 3c grows subexponentially, implying that the total expected running time is also  $O(l^{1+\varepsilon})$ , proving (1).

Claim (2) follows from the same analysis. We have  $\log x_0 = (6 + o(1)) \log l$  and add  $\log 2$  to  $\log x$  in each iteration. Once  $x = x_0$ , it takes more than  $k \log l$  iterations to reach  $\log x > (6 + k) \log l$ . There is a probability of at least  $c$  that the algorithm terminates in each subsequent iteration, yielding the bound in (2).  $\square$

**6.2. CRT computations.** The computations involved in Steps 3, 4b, and 5 of Algorithm 6.1 are described in detail in [54, §6]. We summarize briefly here.

Given  $S = \{p_i\}$ , let  $M = \prod p_i$ ,  $M_i = M/p_i$ , and  $a_i \equiv M_i^{-1} \pmod{p_i}$ . Let  $c$  denote a coefficient of  $\Phi_l \in \mathbf{Z}[X, Y]$ , and let  $c_i \equiv c \pmod{p_i}$  denote the corresponding coefficient of  $\Phi_l \in \mathbf{F}_{p_i}[X, Y]$ . As in [56, §10.3], we can use fast Chinese remaindering

to efficiently compute

$$(15) \quad c \equiv c_i a_i M_i \pmod{M}.$$

Provided that  $M > 2|c|$ , we can then lift the result from  $\mathbf{Z}/M\mathbf{Z}$  to  $\mathbf{Z}$ .

When  $m$  is “large,” by which we mean  $m \geq M > 2|c|$ , we compute  $c \pmod{M}$ , lift to  $\mathbf{Z}$ , and output the integer  $c$  as its representative modulo  $m$ . In this scenario Step 4b simply stores the coefficients  $c_i$  and Step 3 can be deferred to Step 5.

When  $m$  is “small,” by which we mean  $M(\log m) = O(\log^3 l \log \log l)$ , we instead use the explicit CRT modulo  $m$ . Assuming  $M > 4|c|$ , we may apply

$$(16) \quad c \equiv c_i a_i M_i - rM \pmod{m},$$

where  $r$  is the closest integer to  $s = \sum c_i a_i / p_i$ , by [4, Thm. 3.1]. In this scenario, we update the sum  $C = \sum c_i a_i M_i \pmod{m}$  and an approximation to  $s$  in Step 4b as each  $c_i$  is computed. This uses  $O(\log m + \log l)$  space per coefficient, rather than the  $O(l \log l)$  space used to compute  $c \pmod{M}$ . The postcomputation in Step 5 determines  $r$  from the approximation to  $s$  and computes  $c \pmod{m}$  via (16).

When  $m$  is neither small nor large, a hybrid approach is used, see [54, §6.3].

**6.3. Computing  $\Phi_l(X, Y) \pmod{p}$ .** An overview of Algorithm 2.1 was given in the introduction, we now fill in the details.

**Algorithm 2.1.** *Let  $l$ ,  $p$ , and  $\mathcal{O}$  be as in Theorem 4.1, with  $h(\mathcal{O}) \geq l + 2$ , and let  $R = \mathbf{Z} + l\mathcal{O}$ . Given  $H_{\mathcal{O}} \in \mathbf{Z}[X]$ , compute  $\Phi_l \in \mathbf{F}_p[X, Y]$  as follows:*

1. Compute the presentations  $\alpha$  of  $\text{cl}(\mathcal{O})$  and  $\alpha'$  of  $\text{cl}(R)$  suitable for  $p$ .
2. Find a root of  $j_0$  of  $H_{\mathcal{O}}(X)$  over  $\mathbf{F}_p$ .
3. Use  $\alpha$  to enumerate  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$  from  $j_0$  and identify the  $l$ -isogeny cycles.
4. For distinct  $j_0, \dots, j_{l+1} \in \text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$ :
  - a. Construct a curve  $E_i$  with  $j(E_i) = j_i$  such that  $l$  divides  $\#E_i(\mathbf{F}_p)$ .
  - b. Generate a random point  $P \in E_i(\mathbf{F}_p)$  of order  $l$ .
  - c. Use  $E_i$  and  $P$  to compute an  $l$ -isogenous curve  $E'_i/\mathbf{F}_p$  via Algorithm 6.4.
  - d. If  $j(E'_i) \notin \text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$  then set  $j'_i \leftarrow (E'_i)$ , otherwise return to Step 3b.
5. Use  $\alpha'$  to enumerate  $\text{Ell}_R(\mathbf{F}_p)$  from  $j'_0$  and identify the  $l^2$ -isogeny cycles.
6. For  $i$  from 0 to  $l + 1$ :
  - a. Let  $j_{i0}, \dots, j_{il}$  consist of the neighbors of  $j_i$  in its  $l$ -isogeny cycle in  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$  together with the  $l^2$ -isogeny cycle of  $\text{Ell}_R(\mathbf{F}_p)$  containing  $j'_i$ .
  - b. Compute  $\Phi_l(X, j_i) = \sum_k a_{ik} X^k$  as the product  $\prod_k (X - j_{ik})$ .
7. For  $k$  from 0 to  $l + 1$ :
  - a. Interpolate  $\phi_k \in \mathbf{F}_p[Y]$  with  $\deg \phi_k \leq l + 1$  satisfying  $\phi_k(j_i) = a_{ik}$ .
8. Output  $\Phi_l(X, Y) = \sum_k \phi_k(Y) X^k$ .

Steps 2, 6, and 7 involve standard computations with polynomials over finite fields, as described in [56], for example. Step 1 is addressed in Section 5.1, and Steps 3 and 5 are the topic of Section 5.3. Only Step 4 merits further discussion here.

The existence of the curve  $E_i$  constructed in Step 4a is guaranteed by Theorem 4.1. The trace of Frobenius  $t$  of the desired curve is uniquely determined by the norm equation for  $p$  and the constraint  $t \equiv 2 \pmod{l}$ , as in (12). With  $k = j_i/(1728 - j_i)$ , the curve  $E/\mathbf{F}_p$  defined by  $y^2 = x^3 + 3kx + 2k$  has  $j(E) = j_i$ , and we may determine whether it is  $E$  or its quadratic twist that has trace  $t$  by attempting to generate a point of order  $l$  on both curves in Step 4b.



To obtain a point  $P$  of order  $l$ , we generate a random point  $Q$  uniformly distributed over  $E_i(\mathbf{F}_p)$ , compute the scalar multiple  $P = nQ$ , where  $n = (p+1-t)/l$ , and then check that  $P \neq 0$ . This will be true with probability  $1 - 1/l^2$ , since Theorem 4.1 implies that the Sylow  $l$ -subgroup of  $E_i(\mathbf{F}_p)$  is  $E_i[l] \cong \mathbf{Z}/l\mathbf{Z} \times \mathbf{Z}/l\mathbf{Z}$ . Thus we expect to succeed within  $1 + O(1/l^2)$  attempts, and the expected cost of generating  $P$  is  $O(\log p)$  operations in  $\mathbf{F}_p$ .

Note that  $E_i(\mathbf{F}_p)$  contains  $l+1$  distinct subgroups of order  $l$ , each corresponding to a distinct  $l$ -isogenous  $j$ -invariant. At most 2 of these lie in  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$ . Thus in Step 4d we have  $j(E'_i) \notin \text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$  with probability at least  $1 - 2/(l+1)$  and expect to need  $1 + O(1/l)$  random points  $P$  to obtain such an  $E'_i$ . The curve  $E'_i$  is the image of an  $l$ -isogeny whose kernel is generated by  $P$ , obtained via Algorithm 6.4.

**6.4. Isogenies from subgroups.** Let  $E/\mathbf{F}_p$  be an elliptic curve. Given a cyclic subgroup  $H \subseteq E(\overline{\mathbf{F}}_p)$ , Vélu's formulas construct an isogeny  $E \rightarrow E'$  with  $H$  as its kernel [55]. In our setting  $H$  is actually generated by an  $\mathbf{F}_p$ -rational  $l$ -torsion point, allowing us to work in  $\mathbf{F}_p$  rather than an extension field. Additionally, the order  $l$  of  $H$  is odd and  $p > 3$ , allowing us to simplify the formulas.

**Algorithm 6.4.** Let  $l > 2$  and  $p > 3$  be primes, let  $E/\mathbf{F}_p$  be an elliptic curve defined by  $y^2 = x^3 + Ax + B$ , and let  $P = (P_x, P_y)$  be a point on  $E(\mathbf{F}_p)$  of order  $l$ . Compute the image  $E'/\mathbf{F}_p$  of the  $l$ -isogeny with kernel  $H = \langle P \rangle$  as follows:

1. Set  $t \leftarrow 0$ ,  $w \leftarrow 0$ , and  $Q \leftarrow P$ .
2. Repeat  $(l-1)/2$  times:
  - a. Set  $s \leftarrow 6Q_x^2 + 2A$ , and then set  $u \leftarrow 4Q_y^2 + sQ_x$ .
  - b. Set  $t \leftarrow t + s$ ,  $w \leftarrow w + u$ , and  $Q \leftarrow Q + P$ .
3. Set  $A' = A - 5t$  and  $B' = B - 7w$ .
4. Output the curve  $E'/\mathbf{F}_p$  defined by  $y^2 = x^3 + A'x + B'$ .

The addition  $Q + P$  in Step 2b is performed using the group operation in  $E(\mathbf{F}_p)$ . The complexity of Algorithm 6.4 is  $O(l)$  operations in  $\mathbf{F}_p$ .

**6.5. Complexity analysis.** We first bound the complexity of Algorithm 2.1, as used by Algorithm 6.1.

**Lemma 6.5.** *Let  $\mathcal{F}$  be a suitable family of orders that satisfies the condition of Theorem 5.1. For an odd prime  $l$ , let  $\mathcal{O} = \mathcal{F}(l)$  and let  $D = \text{disc}(\mathcal{O})$ . Let  $p$  be a prime in the set  $S$  selected by Algorithm 6.2 on input  $l$  and  $D$ . Assuming the GRH, the expected running time of Algorithm 2.1 is  $O(l^2(\log p)^3 \log \log p)$ .*

*Proof.* We note that  $p = t^2 - v^2 l^2 D > l^4$ , thus  $\log l < \log p$ , and recall that the bit-complexity of multiplying two polynomials of degree  $O(l)$  in  $\mathbf{F}_p[X]$  may be bounded by  $O(M(l \log p))$ , using Kronecker substitution, see Corollaries 8.28 and 9.8 of [56].

In the analysis below we use  $O(M(l \log p)) = O(l(\log p)^2 \log \log p)$ , via the bound  $M(n) = O(n \log n \log \log n)$  for fast multiplication [50]. When computing the cost of multiplications in  $\mathbf{F}_p$  we use the weaker bound  $M(\log p) = O((\log p)^2 / (\log \log p)^c)$ , where  $c$  is any constant, which is more convenient and does not change the overall bound. We bound the cost of inversions in  $\mathbf{F}_p$  by  $O(M(\log p) \log \log p) = O((\log p)^2)$ , via [56, Cor. 11.10].

We now bound the (expected) cost of each step in Algorithm 2.1:

1. We have  $h(\mathcal{O}) < h(R) = O(l^2)$ . As described in Section 5.1, the cost of computing the presentations  $\alpha$  and  $\alpha'$  is  $O(l^2)$  operations in  $\text{cl}(\mathcal{O})$  and  $\text{cl}(R)$ . Using binary quadratic forms, each group operation has complexity  $O((\log l)^2)$ , by [5], yielding an  $O(l^2(\log l)^2) = O(l^2(\log p)^2)$  bound on Step 1.
2. Using Berlekamp's probabilistic root-finding algorithm [3, §7] with a fast GCD computation [56, Alg. 11.4], the expected time to find a single root of  $H_D \in \mathbf{F}_p[X]$  may be bounded by  $O(M(l)(\log l + \log p))$  operations in  $\mathbf{F}_p$ , since  $\deg H_D = O(l)$ . This implies an  $O(l(\log p)^3)$  bound on Step 2.
3. For  $p \in S$  we have  $\omega(v) \leq 2 \log \log p$ , yielding an  $O(\log \log p \log \log \log p)$  bound on  $\max l(\alpha)$ , by Theorem 5.1. From Lemma 4.5,  $O((\log \log p)^2 \log p)$  operations in  $\mathbf{F}_p$  suffice to compute the action of any element of  $\alpha$ . We obtain an  $O(l(\log p)^3)$  bound on the time to enumerate  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$ , which dominates the  $O(l \log l)$  time to identify the  $l$ -isogeny cycles.
4. From the discussion in Section 6.3 and the complexity of Algorithm 6.4, we expect to use  $O(l^2 + l \log p)$  operations in  $\mathbf{F}_p$  during Step 4. This yields an  $O(l^2(\log p)^2 + l(\log p)^3)$  bound.
5. Recall that the surjective map  $\varphi : \text{cl}(R) \rightarrow \text{cl}(\mathcal{O})$  in (3) preserves the norms of representative ideals. The subgroup of  $\text{cl}(R)$  generated by invertible  $R$ -ideals with norms in  $l(\alpha)$  contains  $\varphi^{-1}(\text{cl}(\mathcal{O}))$ . It follows that the elements of  $\alpha'$  with norm at most  $\max l(\alpha)$  generate a subgroup of size at least  $h(\mathcal{O}) > l$ . All but  $O(l)$  of the  $O(l^2)$  steps taken when enumerating  $\text{Ell}_R(\mathbf{F}_p)$  involve these elements, and, as in Step 3, we obtain a total cost of  $O(l^2(\log p)^3)$  for these steps. Assuming the GRH, the remaining elements of  $\alpha'$  all have norm  $O((\log |D|)^2) = O((\log l)^2)$ , by [1], yielding a total cost of  $O(l(\log l)^4(\log p)^3)$  for these steps, via Lemma 4.5. Thus the expected time to enumerate  $\text{Ell}_R(\mathbf{F}_p)$  is  $O(l^2(\log p)^3)$ , which dominates the  $O(l^2 \log l)$  time to identify the  $l^2$ -isogeny cycles.
6. Using a product tree we may compute  $\prod_k (X - j_{ik})$  in time  $O(M(l \log p) \log l)$ , yielding a total cost of  $O(l^2(\log p)^3 \log \log p)$  for Step 6.
7. Using a product tree and fast interpolation [56, Alg. 10.11], we also obtain a cost of  $O(l^2(\log p)^3 \log \log p)$  for Step 7. Here we use the  $O(M(l \log p))$  bit-complexity of polynomial multiplication in  $\mathbf{F}_p[X]$  to bound the cost at each level, rather than using the bound in [56, Cor. 10.12].

The bound  $O(l^2(\log p)^3 \log \log p)$  applies to every step, completing the proof.  $\square$

We are now ready to prove our main theorem, which bounds the complexity of using Algorithm 6.1 to compute  $\Phi_l \bmod m$ , where  $l$  is an odd prime and  $m$  is any positive integer. Recall that that the algorithm must be given a suitable family of orders  $\mathcal{F}$ , as defined in Definition 4.2, and to prove our complexity bound we additionally require that  $\mathcal{F}$  satisfy the property given in Theorem 5.1. Example 4.3 provides one such  $\mathcal{F}$ , and there are many others that can be efficiently computed and may yield better performance, as discussed in Section 6.6.

**Theorem 1.** *Let  $\mathcal{F}$  be a suitable family of orders that satisfies the condition of Theorem 5.1. Let  $l$  be an odd prime and let  $m \in \mathbf{Z}_{>0}$ . Given inputs  $l$ ,  $m$ , and  $\mathcal{O} = \mathcal{F}(l)$ , Algorithm 6.1 correctly computes  $\Phi_l \in (\mathbf{Z}/m\mathbf{Z})[X, Y]$ . Under the GRH, its expected running time is*

$$O(l^3 \log^3 l \log \log l),$$

using  $O(l^2 \log lm)$  expected space.

*Proof.* We first argue correctness. By Lemma 6.3, Algorithm 6.2 obtains a set of primes  $S$  that satisfy Theorem 4.1, with  $\prod_{p \in S} p > 4|c|$ , for every coefficient  $c$  of  $\Phi_l$ . We now claim that for  $p \in S$ , Algorithm 2.1 obtains, for each of  $l + 2$  distinct  $j$ -invariants  $j_i$ , a list of  $l + 1$  distinct  $j$ -invariants  $j_{ik}$  of  $l$ -isogenous curves. Granting the claim, we may invoke standard properties of  $\Phi_l$  to show that Algorithm correctly interpolates  $\Phi_l \in \mathbf{F}_p[X, Y]$ , see [57, Thm. 12.19] and [44, Thm. 5.3], for example. The correctness of Algorithm 6.1 then follows from the CRT and/or the explicit CRT mod  $m$ , via [4, Thm. 3.1], as described in Section 6.2.

As usual, let  $\mathcal{O} = \mathcal{F}(l)$  have fraction field  $K$ , and let  $R = \mathbf{Z} + l\mathcal{O}$ . The claim above rests on three facts: (1) the explicit CM-action described in Section 4.3 is correct, (2) any two  $l^2$ -isogenous elements of  $\text{Ell}_R(\mathbf{F}_p)$  must be  $l$ -isogenous to exactly one and the same element of  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$ , and (3) each  $l^2$  isogeny cycle in  $R$  contains exactly  $l - \left(\frac{d_K}{l}\right)$  elements. We note that (1) follows from the theory of complex multiplication and the properties of  $\Phi_{l_0}$  guaranteed by [57, Thm. 12.19], (2) follows from the  $l$ -volcano structure, as shown by [30, §2.2] and [42, Prop. 23], and (3) is explicitly proven in Lemma 3.1.

We now assume the GRH and bound the complexity of Algorithm 6.1. Lemma 6.3 shows that the expected size of the largest  $p \in S$  is  $O(\log l)$ , and we have  $\#S = O(l)$ . Applying [54, §6] yields an  $O(l^2 \log lm)$  space bound for  $m \in \mathbf{Z}_{>0}$ .

By [54, Thm. 1], the expected time to compute  $H_{\mathcal{O}}$  in Step 1 is  $O(l^{2+\varepsilon})$ , and Lemma 6.3 gives an expected time of  $O(l^{1+\varepsilon})$  for Step 2, for any  $\varepsilon \in \mathbf{R}_{>0}$ . Additionally, we have  $\log p > (6 + k) \log l$  for all  $p \in S$  with probability approaching 1 exponentially as  $k$  increases. The time complexity of all remaining steps in Algorithm 6.1, including calls to Algorithm 2.1, depends polynomially on  $\log p$ , hence we may bound the expected running time assuming  $\log p = O(\log l)$ .

Regardless of the exact cutoff used, if  $M(\log m) = O((\log l)^3 \log \log l)$  whenever we consider  $m$  “small”, we may apply the results of [54, §6] to obtain a bound of  $O(l^3 (\log l)^3 \log \log l)$  on the expected time for all CRT computations, for every  $m \in \mathbf{Z}_{>0}$ . Since  $\mathcal{F}$  satisfies the property of Theorem 5.1, we may apply Lemma 6.5 with  $\log p = O(\log l)$  to obtain an  $O(l^2 (\log l)^3 \log \log l)$  bound on the expected time of each call to Algorithm 2.1. Applying  $\#S = O(l)$  completes the proof.  $\square$

**6.6. Selecting a suitable order.** The family of orders used in Theorem 1 suffices to prove the complexity bound, but we can simplify the implementation and improve performance with some additional constraints on the order  $\mathcal{O}$ . Let us fix a bound  $b < l$  (say  $b = 256$ , for large  $l$ ), and a small prime  $l_0 < l$  (typically  $l_0 = 2$ ). As above,  $R$  is the order of index  $l$  in  $\mathcal{O}$ , and  $\mathcal{O}_K$  is the maximal order. We seek an order  $\mathcal{O}$  for which the following hold:

- (1) The conductor of  $\mathcal{O}$  is  $b$ -smooth,  $h(\mathcal{O}_K) \leq b$ , and  $h(\mathcal{O}) \geq l + 2$ .
- (2) The groups  $\text{cl}(\mathcal{O})$  and  $\text{cl}(R)$  are either generated by a single ideal with norm  $l_0$ , or by two ideals with norms  $l_0$  and  $l_1$ , where  $l_1 \leq b$  is ramified.

The first condition ensures that  $\mathcal{O}$  is suitable for  $l$  and allows us to obtain a root of  $H_{\mathcal{O}}(X)$  using only polynomials of degree at most  $b$ . This is accomplished by finding a root of  $H_{\mathcal{O}_K}(X)$  and descending to the proper level of the  $l'$ -isogeny volcano for each prime  $l' \leq b$  dividing the conductor of  $\mathcal{O}$ , as in [54, §4.1]. The second condition allows us to realize the torsors for  $\text{cl}(\mathcal{O})$  and  $\text{cl}(R)$  either by walking a single  $l_0$ -isogeny cycle, or by walking two  $l_0$ -isogeny cycles connected by a single  $l_1$ -isogeny. In the latter case we orient the two cycles by computing one

extra  $l_1$ -isogeny. In both cases we avoid the need to ever distinguish the action of an ideal and its inverse, simplifying the computation described in Section 5.3.

Subject to these conditions, we also wish to minimize  $h(\mathcal{O}) \geq l + 2$ . To find such orders we enumerate fundamental discriminants  $d_K < -4$  with  $\left(\frac{d_K}{l_1}\right) = 1$  and  $h(d_K) \leq b$ , and for each  $d_K$  we select  $b$ -smooth integers  $u$  for which  $h(u^2 d_K)$  is slightly greater than  $l + 2$  and test whether condition (2) holds. In practice we are almost always able to obtain  $\mathcal{O}$  with  $h(\mathcal{O})$  within a few percent of  $l + 2$ .

## 7. MODULAR FUNCTIONS OTHER THAN $j$

Let  $g$  be a modular function of level  $N$ , and let  $l \nmid N$  be a prime. We define the *modular polynomial*  $\Phi_l^g$  of level  $l$  for  $g$  as the minimal polynomial of the function  $g(lz)$  over the field  $\mathbf{C}(g)$ . Much of the theory for the classical modular polynomial of the  $j$ -function generalizes to  $g$ . In particular, if the Fourier expansion of  $g$  has integer coefficients then we have  $\Phi_l^g \in \mathbf{Z}(g)[X]$ . The following lemma gives us further information in this case.

**Lemma 7.1.** *Let  $g$  be a modular function and let  $l$  be a prime not dividing the level of  $g$ . Suppose that  $\Phi_l^g$  has integer coefficients. If  $g$  is invariant under the action of either  $S = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \in \mathrm{SL}_2(\mathbf{Z})$  or  $M = \begin{pmatrix} 0 & -l \\ 1 & 0 \end{pmatrix} \in \mathrm{GL}_2(\mathbf{Q})$ , then we have*

$$\Phi_l^g(X, Y) = \Phi_l^g(Y, X).$$

*Proof.* The proof follows the symmetry proof for  $\Phi_l(X, Y)$ , see [44, Thm. 5.3].  $\square$

The polynomial  $\Phi_l^g$  should not be confused with the minimal polynomial of  $g$  as an element of  $\mathbf{C}(j)$ , which we denote  $\Psi^g(X, J)$ . The polynomial  $\Psi^g$  depends only on  $g$ , not  $l$ , and we assume it is known (for our purposes, it effectively defines  $g$ ). Given  $\Psi^g$ , our goal is to efficiently compute  $\Phi_l^g$  for a prime  $l \nmid N$ .

To apply our method we require that  $\Phi_l^g$  have degree  $l + 1$  (in both  $X$  and  $Y$ ). The degree of  $\Phi_l^g$  can be explicitly computed using [12, §5.2], and we note that this degree must be  $l + 1$  when  $\deg_J \Psi^g = 1$  (this applies to the function  $\gamma_2$  and the Weber  $\mathfrak{f}$ -function considered in Sections 7.1 and 7.3), and also when  $\deg_J \Psi^g = 2$  and  $g$  is invariant under the Atkin-Lehner involution (this applies to the various modular functions considered in Section 7.4).

We wish to adapt Algorithm 2.1 to compute  $\Phi_l^g \in \mathbf{F}_p[X]$ . We may then apply Algorithm 6.1 to recover  $\Phi_l^g$  over the integers or modulo some integer  $m$  via the Chinese Remainder Theorem. To simplify matters, we place some additional restrictions on the order  $\mathcal{O}$  that we use in Algorithm 6.1. Specifically, we require that there is a generator  $\tau \in \mathbf{H}$  of  $\mathcal{O} = \mathbf{Z}[\tau]$  with the property that

$$g(\tau) \in K_{\mathcal{O}},$$

where  $K_{\mathcal{O}}$  is the ring class field for the order  $\mathcal{O}$ . We say that  $g$  is a *class invariant* for  $\mathcal{O}$  in this case. If we now take a prime  $p$  that splits completely in  $K_{\mathcal{O}}$  and  $E/\mathbf{F}_p$  an elliptic curve with endomorphism ring  $\mathcal{O}$ , then the polynomial

$$\Psi^g(X, j(E)) \in \mathbf{F}_p[X]$$

has at least one root in  $\mathbf{F}_p$ . Indeed, the value  $h = g(\tau) \bmod \mathfrak{p}$ , for a prime  $\mathfrak{p}|p$  of  $K_{\mathcal{O}}$ , satisfies  $\Psi^g(h, j(E)) = 0$ .

We can analyze *how many roots* the polynomial  $\Psi^g(X, j(E))$  has in  $\mathbf{F}_p$  using a combination of Deuring lifting and Shimura reciprocity. We refer to [10, §6.7] for a detailed description of the techniques involved and only state the result here. Let

$g_i : \mathbf{H} \rightarrow \mathbf{C}$  be the roots of  $\Psi^g(X, j)$ . The functions  $g_i$  are modular of level  $N$ , and they are permuted by the Galois group  $\mathrm{GL}_2(\mathbf{Z}/N\mathbf{Z})$  of the field of all modular functions of level  $N$ . To state our result, we will associate a matrix  $A \in \mathrm{GL}_2(\mathbf{Z}/N\mathbf{Z})$  to the Frobenius morphism of  $E$  as follows. Fix an isomorphism  $\mathrm{End}(E) \xrightarrow{\sim} \mathcal{O}$  and let  $\pi_p \in \mathcal{O}$  be the image of the Frobenius morphism. If  $\pi_p$  has minimal polynomial  $X^2 - tX + p$  of discriminant  $\Delta = t^2 - 4p$ , then we put

$$(17) \quad A = \begin{pmatrix} \frac{t-2}{2} & \frac{\Delta-1}{2} \\ \frac{2}{t+2} & \frac{2}{t-2} \end{pmatrix} \in \mathrm{GL}_2(\mathbf{Z}/N\mathbf{Z}).$$

**Theorem 7.2.** *Let  $g$  be a modular function with the property that  $\Psi^g$  has integer coefficients and is separable modulo a prime  $p$  that does not divide the level of  $N$ . If  $E/\mathbf{F}_p$  is an elliptic curve with endomorphism ring  $\mathcal{O}$  and  $j_0 = j(E)$ , then we have*

$$\#\{x \in \mathbf{F}_p : \Psi^g(x, j_0) = 0\} = \#\{g_i : \Psi^g(g_i, j) = 0 \text{ and } g_i^A = g_i\},$$

where  $A$  is the matrix in (17).

*Proof.* See [10, §6.7]. □

To check if  $g_i^A = g_i$  holds is a standard computation, see [32] for example. Although the matrix  $A$  has norm  $p$  and trace  $t$  and therefore depends on  $p$ , we can often derive a result that merely depends on a congruence condition on  $p \bmod N$ . Lemma 7.3 in Section 7.3 gives an example.

**7.1. Computing modular polynomials for  $\gamma_2$ .** Let  $\gamma_2(z)$  denote the unique cube root of  $j(z)$  that has integral Fourier expansion. It was known to Weber already that  $\gamma_2$  is a modular function of level 3, see [58, §125], and  $\gamma_2$  is a class invariant for  $\mathcal{O}$  whenever  $3 \nmid \mathrm{disc}(\mathcal{O})$ . We have  $\Psi^{\gamma_2}(X, j) = X^3 - j$ , and in this simple case there is no need to apply Theorem 7.2; if we restrict to  $p = 2 \bmod 3$  then every element of  $\mathbf{F}_p$  has a unique cube root. Thus we can compute  $\Phi_l^{\gamma_2}$  with only minor modifications to Algorithm 6.1:

- Use a suitable order  $\mathcal{O}$  with  $3 \nmid \mathrm{disc}(\mathcal{O})$  and select only primes  $p \equiv 2 \pmod{3}$ .
- After Step 5 of Algorithm 2.1, replace each element of  $\mathrm{Ell}_{\mathcal{O}}(\mathbf{F}_p)$  and  $\mathrm{Ell}_R(\mathbf{F}_p)$  with its unique cube root in  $\mathbf{F}_p$ .

These changes suffice, but we can also improve the algorithm's performance.

First, Lemmas 2–3 and Corollary 9 of [13] yield the bound

$$(18) \quad B_l^{\gamma_2} = 2l \log l + 8l$$

on the logarithmic height of  $\Phi_l^{\gamma_2}$  (conjecturally,  $B_l^{\gamma_2} = 2l \log l + 4l$  for all  $l > 60$ , but we do not use this). Thus we can reduce the height bound  $B_l$  in (14) by a factor of approximately 3 when computing  $\Phi_l^{\gamma_2}$ . This reduces the number of primes  $p \in S$ , and the corresponding number of calls Algorithm 6.1 makes to Algorithm 2.1.

Second, we may take advantage of the fact that  $\Phi_l^{\gamma_2}$  is *sparser* than  $\Phi_l$ . As noted in [22, p. 37], the coefficient of  $X^a Y^b$  in  $\Phi_l^{\gamma_2}$  is zero unless

$$(19) \quad a + lb \equiv l + 1 \pmod{3}.$$

The proof of this relation goes back to Weber: the argument given in [58, p. 266] generalizes immediately to  $\gamma_2$ . It allows us to reduce the number of points we use to interpolate  $\Phi_l^{\gamma_2}$  by a factor of approximately 3.

Let  $n = \lceil (l+1)/3 \rceil + 1$ . When selecting a suitable order  $\mathcal{O}$  as in Section 6.6, we now only require that  $h(\mathcal{O}) \geq n$ , and further modify Algorithm 2.1 as follows:

- In Steps 4-6 we construct just  $n$  polynomials  $\Phi_l^{\gamma_2}(X, \sqrt[3]{j_i})$  of degree  $l + 1$ .
- In Step 7 we interpolate  $l + 1$  polynomials  $\phi_k^*$  of degree less than  $n$  by writing  $\phi_k = Y^c \phi_k^*(Y^3)$ , with  $c \in \{0, 1, 2\}$  satisfying  $c + lk \equiv l + 1 \pmod{3}$ .

This reduces the cost of all the significant components of Algorithm 2.1 by a factor of approximately 3. The reduction in the cost of the interpolations in Step 7 is actually greater than this, since its complexity is superlinear in the degree.

The total size of  $\Phi_l^{\gamma_2}$  is approximately 9 times smaller than  $\Phi_l$ , and with the optimizations above, the time to compute it is effectively reduced by the same factor. A small amount of additional time is required to compute the cube roots of the elements in  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$  and  $\text{Ell}_R(\mathbf{F}_p)$ , but even this can be avoided.

Provided we have already computed  $\Phi_{l'}^{\gamma_2}$  for some small values of  $l'$  (specifically, for the primes  $l_0$  and  $l_1$  of Section 6.6), we may use these polynomials to directly enumerate sets  $\text{Ell}_{\mathcal{O}}^{\gamma_2}(\mathbf{F}_p)$  and  $\text{Ell}_R^{\gamma_2}(\mathbf{F}_p)$  containing the cube roots of the elements in  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$  and  $\text{Ell}_R(\mathbf{F}_p)$  respectively. We need only compute the cube roots of  $j_0$  and  $j'_0$  as starting points. This third optimization yields a small but useful improvement in the case of  $\gamma_2$ , and plays a critical role in the examples that follow.

**7.2. Recovering  $\Phi_l$  from  $\Phi_l^{\gamma_2}$ .** Having computed  $\Phi_l^{\gamma_2}$ , we note that  $\Phi_l$  may be computed via [22, Eq. 23]:

$$(20) \quad \Phi_l(X^3, Y^3) = \Phi_l^{\gamma_2}(X, Y)\Phi_l^{\gamma_2}(X, \omega Y)\Phi_l^{\gamma_2}(X, \omega^2 Y),$$

where  $\omega = e^{2\pi i/3}$ . For computation in  $\mathbf{Z}$ , or modulo  $m$ , it is more convenient to express  $\Phi_l^{\gamma_2}$  in terms of polynomials  $P_0, P_1, P_2 \in \mathbf{Z}[X, Y]$  satisfying

$$(21) \quad \Phi_l^{\gamma_2}(X, Y) = P_0(X^3, Y^3)Y^b + P_1(X^3, Y^3)XY + P_2(X^3, Y^3)X^2Y^{2-b},$$

where  $b = 2$  when  $l \equiv 1 \pmod{3}$ , and  $b = 0$  when  $l \equiv 2 \pmod{3}$ . We then have

$$(22) \quad \Phi_l = P_0^3 Y^b + (P_1^3 - 3P_0 P_1 P_2)XY + P_2^3 X^2 Y^{2-b}.$$

Using Kronecker substitution and fast multiplication, it is possible to evaluate (22) in time  $O(l^3(\log l)^{2+\varepsilon})$ , which is asymptotically faster than Algorithm 6.1. This suggests that we might more efficiently compute  $\Phi_l$  by recovering it from  $\Phi_l^{\gamma_2}$ , but we do not find this to be true in practice: it actually takes longer to evaluate (22) than it does to compute  $\Phi_l$  directly. This can be explained by two factors. First, the  $\mathbf{F}_p$ -operations used in Algorithm 2.1 effectively have unit cost for word-size primes, making it faster than Theorem 1 would suggest for all but very large  $l$ . Secondly, the evaluation of (22) becomes extremely memory intensive when  $l$  is large. However, if we are computing  $\Phi_l \pmod{m}$  with  $\log m \ll l \log l$ , then the time to apply (22) modulo  $m$  is negligible. In this situation it is quite advantageous to derive  $\Phi_l \pmod{m}$  from  $\Phi_l^{\gamma_2} \pmod{m}$ , as may be seen in Table 3 of Section 8.

**7.3. Computing modular polynomials for the Weber  $\mathfrak{f}$  function.** We now consider the classical Weber function [58, p. 114] defined by

$$\mathfrak{f}(z) = \zeta_{48}^{-1} \frac{\eta((z+1)/2)}{\eta(z)},$$

where  $\zeta_{48} = e^{\frac{\pi i}{24}}$  and  $\eta(z)$  is the Dedekind eta function. This is a modular function of level 48 that satisfies  $\gamma_2 = (\mathfrak{f}^{24} - 16)/\mathfrak{f}^8$ , see [58, p. 179], thus we have

$$\Psi^{\mathfrak{f}}(X, j) = (X^{24} - 16)^3 - X^{24}j.$$

Asymptotically, we expect to be able to reduce the height bound  $B_l$  by a factor of  $\deg_X \Psi^f / \deg_j \Psi^f = 72$  when computing  $\Phi_l^f$ . One can derive an explicit bound along the lines of (18), but this tends to overestimate the  $O(l)$  term quite significantly, so in practice for large  $l$  we use the heuristic bound

$$(23) \quad B_l^f = \frac{1}{12}l \log l + \frac{1}{5}l \quad (l > 2400),$$

which has been verified for every prime  $l$  between 2400 and 10000. The modular polynomial  $\Phi_l^f$  is also sparse: the coefficient of  $X^a Y^b$  can be nonzero only when

$$(24) \quad la + b \equiv l + 1 \pmod{24},$$

as shown in [58, p. 266]. Thus  $\Phi_l^f$  is roughly  $72 \cdot 24 = 1728$  times smaller than  $\Phi_l$ . By applying the technique described above for  $\gamma_2$ , *mutatis mutandis*, we can actually compute  $\Phi_l^f$  more than 1728 times faster than  $\Phi_l$  for large values of  $l$ , as may be seen in Table 2 of Section 8. When applying Algorithm 6.1, we now insist that the order  $\mathcal{O}$  have discriminant  $D \equiv 1 \pmod{8}$  and  $3 \nmid D$ , since the Weber function yields class invariants in (at least) this case, see [32], for example.

Since  $-f$  will also yield class invariants for  $\mathcal{O}$ , the polynomial  $\Psi^f(X, j_0)$  will always have at least *two* roots in  $\mathbf{F}_p$ . The following lemma tells us that we can impose a congruence condition on  $p$  to ensure that we have exactly two roots.

**Lemma 7.3.** *Let  $p \equiv 11 \pmod{12}$  be prime and let  $j_0$  be the  $j$ -invariant of an elliptic curve  $E/\mathbf{F}_p$  with  $\text{End}(E)$  isomorphic to an imaginary quadratic order  $\mathcal{O}$  with discriminant  $D \equiv 1 \pmod{8}$  and  $3 \nmid D$ . Then  $\Psi^f(X, j_0) \in \mathbf{F}_p[X]$  has exactly two roots in  $\mathbf{F}_p$ , and these are of the form  $x_0$  and  $-x_0$ .*

Note that if the lemma applies to  $\mathcal{O}$ , it also applies to the order  $R$  of index  $l$  in  $\mathcal{O}$ .

*Proof.* We only have to apply Theorem 7.2. The action of  $A$  on the roots of  $\Psi^f(X, j_0)$  is computed in [10, §6.7], and this yields the lemma.  $\square$

Given a  $j$ -invariant  $j_0 \in \mathbf{F}_p$  that corresponds to  $j(\tau_0) \in K_{\mathcal{O}}$ , we cannot readily determine which of the roots  $x_0$  and  $-x_0$  of  $\Psi^f(X, j_0)$  actually corresponds to  $f(\tau_0)$ . The functions  $f$  and  $-f$  yield distinct class invariants, but they share the same modular polynomials, since  $\Phi_l^f(X, Y) = \Phi_l^f(-X, -Y) = \Phi_l^{-f}(X, Y)$ , by (24).

Thus for the initial  $j_0$  obtained in Step 2 of Algorithm 2.1, it does not matter whether we pick  $x_0$  or  $-x_0$  as a root of  $\Psi^f(X, j_0)$ , and we need not be concerned with making a consistent choice for each prime  $p$ . However it is critical that while computing  $\Phi_l^f \pmod{p}$  we make a consistent choice of sign for each  $j$ -invariant we convert to an “ $f$ -invariant” (a root of  $\Psi^f(X, j_i) \pmod{p}$ ). This makes it impractical to enumerate  $j$ -invariants and convert them *en masse*. Instead, as described for  $\gamma_2$  above, we use modular polynomials  $\Phi_{l'}^f$  for small  $l'$  to enumerate sets  $\text{Ell}_{\mathcal{O}}^f(\mathbf{F}_p)$  and  $\text{Ell}_R^f(\mathbf{F}_p)$  from starting points  $x_0$  and  $x'_0$  satisfying  $\Psi^f(x_0, j_0) = \Psi^f(x'_0, j'_0) = 0$ . This ensures that signs are chosen consistently within each of these sets; we only need to check that the sign choices for the two sets are consistent with each other.

To do so, we use the fact that the coefficient of  $X^l Y^l$  in  $\Phi_l^f(X, Y)$  is  $-1$ . This is shown for  $\Phi_l$  in [58, §69], and the same argument applies to  $\Phi_l^f$ . We modify Algorithm 2.1 to compute the coefficient of  $X^l Y^l$  in  $\Phi_l^f \pmod{p}$  in between Steps 5 and 6. We do this twice, switching the signs in  $\text{Ell}_{\mathcal{O}}^f(\mathbf{F}_p)$  the second time, and expect exactly one of these computations to yield  $-1$ , thereby determining a consistent

choice of signs. This test should be regarded as a heuristic, since we do not rule out the possibility that both choices produce  $-1$ . However, in the course of extensive testing this has never happened, and we suspect that it cannot. If it does occur, the algorithm can detect this and simply choose a different prime  $p$ .

**7.4. Eta quotients and Atkin modular functions.** For a prime  $N$ , let

$$f_N(z) = N^{s/2} \left( \frac{\eta(Nz)}{\eta(z)} \right)^s,$$

where  $s = 24/\gcd(12, N-1)$ . These are modular functions of level  $N$ , and the polynomials  $\Psi_N = \Psi^{f_N}$  that relate  $f_N$  to  $j$  are sometimes called canonical modular polynomials [19, p. 418]. The functions  $f_N$  are closely related to the functions  $\mathfrak{w}_N^s$  considered in [25], and in fact  $\Psi^{f_N} = \Psi^{\mathfrak{w}_N^s}$ , so what follows applies to both. When  $N$  is 2, 3, 5, 7, or 13, we have  $\deg_j \Psi_N = 1$  and can adapt Algorithm 6.1 to compute polynomials  $\Phi_l^{f_N}$  for odd primes  $l \nmid N$ . We assume here that  $N$  is also odd.

We have  $\deg_X \Psi_N = N+1$ , hence we can reduce the height bound  $B_l$  by a factor of approximately  $N+1$ . When selecting a suitable order  $\mathcal{O}$ , we require that  $N$  is prime to the conductor and splits into prime ideals that are distinct in  $\text{cl}(\mathcal{O})$ . This assumption is stronger than we need, but it simplifies the implementation. For the primes  $p \in S$  we require that  $N$  is prime to  $v$ , where  $4p = t^2 - v^2 \text{disc}(\mathcal{O})$ .

As shown in [48], the polynomial  $\Psi_N(X, j_0) \bmod p$  has the same splitting type as  $\Phi_N(X, j_0) \bmod p$ . In particular, for  $j_0 \in \text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$  (or  $j_0 \in \text{Ell}_R(\mathbf{F}_p)$ ) it has exactly two roots, say  $x_1$ , and  $x_2$ . These correspond to  $N$ -isogenies as follows: the  $j$ -invariants  $j_1$  and  $j_2$  of the two elliptic curves that are  $N$ -isogenous to  $j_0$  are uniquely determined by the relations  $\Psi_N(N^s/x_1, j_1) = 0$  and  $\Psi_N(N^s/x_2, j_2) = 0$ . Here the transformation  $x \mapsto N^s/x$  realizes the Atkin-Lehner involution on  $f_N(z)$ .

Starting points  $x_0$  and  $x'_0$  corresponding to  $j_0$  and  $j'_0$  are chosen as follows. Let  $x'_0$  be a root of  $\Psi_N(X, j'_0)$ , chosen arbitrarily, and let  $j'_1$  be determined by  $\Psi_N(N^s/x'_0, j'_1) = 0$ . We then use Vélú's formulas to obtain the  $j$ -invariant  $j_1$  of the elliptic curve that is  $l$ -isogenous to  $j'_1$  (there is exactly one and it lies in  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$ , since  $j'_1 \in \text{Ell}_R(\mathbf{F}_p)$  is on the floor of its  $l$ -volcano). Finally,  $x_0$  is uniquely determined by the constraints  $\Psi_N(x_0, j_0) = 0$  and  $\Psi_N(N^s/x_0, j_1) = 0$ .

We next consider double eta-quotients [26, 27] of composite level  $N = p_1 p_2$ :

$$\mathfrak{w}_{p_1, p_2}^s(z) = \left( \frac{\eta(\frac{z}{p_1})\eta(\frac{z}{p_2})}{\eta(\frac{z}{p_1 p_2})\eta(z)} \right)^s,$$

where  $p_1 \neq p_2$  are primes and  $s = 24/\gcd(24, (p_1-1)(p_2-1))$ . For  $(p_1, p_2)$  in

$$\{(2, 3), (2, 5), (2, 7), (2, 13), (3, 5), (3, 7), (3, 13), (5, 7)\},$$

the polynomial  $\Psi_{p_1, p_2} = \Psi^{\mathfrak{w}_{p_1, p_2}^s}$  has degree 2 in  $j$  and we can compute  $\Phi_l^{\mathfrak{w}_{p_1, p_2}^s}$  for odd primes  $l \nmid N$ . Our restrictions on  $\mathcal{O}$  are analogous to those for  $f_N$  or  $\mathfrak{w}_N^s$ : we require that  $N$  is prime to the conductor and that both  $p_1$  and  $p_2$  split into distinct prime ideals in  $\text{cl}(\mathcal{O})$ . Our requirements for  $p \in S$  are as above. We can reduce the height bound  $B_l$  by a factor of approximately  $(p_1+1)(p_2+1)/2$ .

With the double eta-quotients, the polynomial  $\Psi_{p_1, p_2}(X, j_0)$  has four roots, corresponding to four distinct isogenies of (composite) degree  $N$ . Each root  $x_i$  uniquely determines the  $j$ -invariant of a curve  $N$ -isogenous to  $E/\mathbf{F}_p$  as the unique root of  $\Psi_{p_1, p_2}(x_i, J)/(J - j_0) \in \mathbf{F}_p[J]$ . The double eta-quotients are invariant under the



Atkin-Lehner involution, so we need not transform  $x_i$ . With this understanding, the procedure for selecting  $x_0$  and  $x'_0$  is as above.

In some cases one can obtain smaller modular polynomials by considering suitable roots of the functions defined above. For example, a sixth root of  $f_3$  also yields class invariants (this is shown for  $\mathfrak{w}_3^2$  in [25]), and the corresponding modular polynomials are sparser and of lower height (by a factor of 6).

Our algorithm also applies to the Atkin modular functions, which we denote  $A_N$ . These are (optimal) modular functions for  $X_0^+(N)$  invariant under the Atkin-Lehner involution, see [22, 47] for further details. The polynomials  $\Psi^{A_N}$  are known as Atkin modular polynomials, and are available in computer algebra systems such as Magma [15] and Sage [52]. For primes  $N < 32$ , and also  $N$  in the set  $\{41, 47, 59, 71\}$ , we have  $\deg_j \Psi^{A_N} = 2$  and can compute polynomials  $\Phi_l^{A_N}$  for odd primes  $l \neq N$ . This is done in essentially the same way as with the double eta-quotients, except that now  $N$  is prime and  $\Psi^{A_N}(X, j_0)$  has just two roots, rather than four. For these  $A_N$ , the height bound can be reduced by a factor of approximately  $(N + 1)/2$ .

Finally, we note an alternative approach applicable to both eta-quotients and the Atkin modular functions. If we choose  $\mathcal{O}$  so that the prime factors of  $N$  are all ramified, then there is actually a unique  $x_0 \in \mathbf{F}_p$  corresponding to each  $j_0$  in  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$  and  $\text{Ell}_R(\mathbf{F}_p)$ , that is,  $\Psi^g(X, j_0)$  has exactly one root in  $\mathbf{F}_p$ . In this scenario we can simply enumerate  $j$ -invariants as usual and then replace each  $j_i$  with a corresponding  $x_i$ . This is not as efficient and places stricter requirements on  $\mathcal{O}$ , but it allows us to compute  $\Phi_l^g$  without needing to know  $\Phi_{l'}^g$  for any  $l'$ . This provides a convenient way to “bootstrap” the process. In fact all of the modular polynomials  $\Phi_l^g$  we have considered can eventually be obtained via Algorithm 6.1, starting from the polynomials  $\Psi^g$  and  $\Phi_2$ .

## 8. COMPUTATIONAL RESULTS

We have applied our algorithm to compute polynomials  $\Phi_l^g$  for all the modular functions discussed in Section 7 and every applicable  $l$  up to 1000. For the functions  $j$ ,  $\gamma_2$ , and  $\mathfrak{f}$  we have gone further, and present details of these computations here.

**8.1. Implementation.** The algorithms described in this paper were implemented using the GNU C/C++ compiler [29] and the GMP library [33] on a 64-bit Linux platform. Multiplication of large polynomials is handled by the `zn_poly` library developed by Harvey [36, 35].

The hardware platform included four 3.0 GHz AMD Phenom II processors, each with four cores and 8GB of memory. Up to 16 cores were used in the larger tests, with essentially linear speedup. For consistency we report total CPU times, noting that in a multi-threaded implementation, disk and network I/O can be overlapped with CPU activity so that all computations are CPU bound.

As a practical optimization, we do not use the Hilbert class polynomial  $H_{\mathcal{O}}$  in Step 1 of Algorithm 6.1. Instead, we compute the minimal polynomial of some more favorable class invariant, as described in [28], which is then used to obtain a  $j$ -invariant. Additionally, as noted in Section 6.6, it suffices to compute a class polynomial for the maximal order containing  $\mathcal{O}$ . With these optimizations the time spent computing class polynomials is completely negligible (well under one second).

Another important optimization is the use of polynomial gcds to accelerate root-finding when walking paths in the isogeny graph, a technique developed in [28, §2]. This greatly accelerates the enumeration of the sets  $\text{Ell}_{\mathcal{O}}(\mathbf{F}_p)$  and  $\text{Ell}_R(\mathbf{F}_p)$  in Steps

$l$	$ D $	$n$	$B_l$	$b_l$	size (MB)	time (s)	MB/s
101	216407	184	6511	5751	2.65	2.25	1.18
211	393047	369	14949	13359	27.6	14.4	1.92
307	837407	531	22748	20483	90.5	51.0	1.78
401	626431	725	30640	27642	211	130	1.62
503	3076175	870	39421	35686	431	264	1.63
601	461351	1011	48027	43542	755	485	1.56
701	1254871	1229	56953	51731	1227	863	1.42
809	916599	1376	66731	60743	1926	1410	1.37
907	986855	1517	75712	69017	2759	2010	1.37
1009	2871983	1728	85157	77653	3857	2910	1.32
2003	91696103	3410	180941	166095	33120	31800	1.04
3001	248329639	5122	281635	259272	117256	143000	0.82
4001	72135279	6939	385300	355707	287783	363000	0.79
5003	67243191	8373	491355	454429	577740	749000	0.77

TABLE 1. Computations of  $\Phi_l$  over  $\mathbf{Z}$ .

3 and 5 of Algorithm 6.1. As a result, most of the computation (typically over 75%) is spent interpolating polynomials in Steps 6 and 7.

**8.2. Computations over  $\mathbf{Z}$ .** Tables 1 and 2 provide performance data for computations of  $\Phi_l$  and  $\Phi_l^f$  using Algorithm 6.1. For each  $l$  we list:

- The discriminant  $D$  of the suitable order  $\mathcal{O}$ .
- The number of CRT primes  $n = \#S$  used.
- The height bound  $B_l$  in bits and the actual bit-size  $b_l$  of the largest coefficient.
- The total size of  $\Phi_l$  (resp.  $\Phi_l^f$ ) in megabytes (1MB =  $10^6$  bytes), computed as the sum of the coefficient sizes, with symmetric terms counted only once.
- The total CPU time, in seconds. This includes the time to select  $\mathcal{O}$ .
- The throughput, defined as the total size divided by the total CPU time.

In the last column of Table 1 one can see the quasilinear performance of Algorithm 6.1 as a function of the size of  $\Phi_l$ , and the constant factors appear to be advantageous relative to other algorithms. For example, computing  $\Phi_{1009}$  with the evaluation/interpolation algorithm of [24] uses approximately 100000 CPU seconds (scaled to our hardware platform), while Algorithm 6.1 needs less than 3000.

The first five rows of Table 2 may be compared to the corresponding rows of Table 1 to see the performance advantage gained when computing modular polynomials for the Weber  $f$  function rather than  $j$ . As expected, these polynomials are approximately 1728 times smaller, and the speedup achieved by Algorithm 6.1 is even better; we already achieve a speedup of around 1800 when  $l = 1009$ , and this increases to over 3000 when  $l = 5003$ . This can be explained by the superlinear complexity of interpolation, as well as the superior cache utilization achieved by condensing the sparse coefficients of  $\Phi_l^f$ , as described in Section 7.1.

As noted in Section 7.3, we used a heuristic height bound for the computations in Table 2. The gap between the values of  $b_l$  and  $B_l$  in each case gives us high confidence in the results (the probability of this occurring by chance is negligible).

$l$	$ D $	$n$	$B_l$	$b_l$	size (MB)	time (s)	MB/s
1009	1391	33	1275	1099	2.34	1.59	1.47
2003	37231	58	2542	2271	19.5	10.7	1.81
3001	88879	88	3822	3611	69.6	47.7	1.46
4001	53191	112	5201	4801	167	116	1.45
5003	30959	136	6613	6228	339	241	1.41
6007	463039	170	8052	7530	595	493	1.21
7001	150631	192	9496	8876	957	701	1.37
8009	315031	220	10979	10292	1453	1200	1.21
9001	179159	240	12453	11974	2123	1790	1.18
10009	207919	265	13964	13453	2953	2630	1.12
20011	1114879	537	29485	27860	24942	27600	0.90
30011	2890639	795	45649	43304	87660	123000	0.71
40009	22309439	1032	62210	59439	214273	335000	0.64
50021	37016119	1316	79116	78077	508571	677000	0.75
60013	27334823	1594	96165	91733	747563	1150000	0.65

TABLE 2. Computations of  $\Phi_l^f$  over  $\mathbf{Z}$ .

**8.3. Computations modulo  $m$ .** Table 3 gives timings for computations of  $\Phi_l$  modulo 256-bit and 1024-bit primes  $m$ . The values of  $m$  are arbitrary, and, in particular, they are not of a form suitable for direct computation with Algorithm 2.1. Instead, Algorithm 6.1 derives  $\Phi_l \bmod m$  from the computations of  $\Phi_l \bmod p$ , for  $p \in S$ , using the explicit CRT. The same set  $S$  is used as when computing  $\Phi_l$  over  $\mathbf{Z}$ , so the running time is largely independent of  $m$ , but using the explicit CRT yields a noticeable speedup when  $\log m$  is significantly smaller than  $6l \log l$ . For example, when  $l = 1009$  it takes approximately 2300 seconds to compute  $\Phi_l \bmod m$ , for the  $m$  listed in Table 3, versus about 2900 seconds to compute  $\Phi_l$  over  $\mathbf{Z}$ .

In addition to computing  $\Phi_l \bmod m$  directly, we may also obtain  $\Phi_l \bmod m$  by computing  $\Phi_l^{\gamma^2} \bmod m$  and applying (22), as discussed in Section 7.1. The time to compute  $\Phi_l^{\gamma^2} \bmod m$  is essentially independent of  $m$ , but the time to apply (22) is not. Even so, for the 256-bit and 1024-bit  $m$  that we used, computing  $\Phi_l \bmod m$  in this fashion is much faster than computing  $\Phi_l \bmod m$  directly; for  $l = 1009$  we achieve times of 223 and 403 seconds, respectively. As with  $\Phi_l^f$ , this speedup improves superlinearly, and for large  $l$  it exceeds the expected factor of 9.

When computing  $\Phi_l^{\gamma^2} \bmod m$  we used the height bound  $B_l^{\gamma^2} = 2l \log l + 8l$  given by (18). The timings in Table 3 would be further improved if the heuristic bound  $B_l^{\gamma^2} = 2l \log l + 4l$  were used instead.

The computations listed in Tables 1 and 2 were practically limited by space, not time. The largest computations took only a day or two when run on 16 cores, but required nearly a terabyte of disk storage. However when computing  $\Phi_l \bmod m$ , we can handle larger values of  $l$  without using an excessive amount of space. When  $l = 20011$ , for example, the total size of  $\Phi_l$  is over 30 terabytes, but we are able to compute  $\Phi_l$  modulo a 256-bit integer  $m$  using less than 10 gigabytes.

$l$	$m = 2^{256} - 189$			$m = 2^{1024} - 105$		
	$\Phi_l$	$\Phi_l^{\gamma^2}$	$\Phi_l^*$	$\Phi_l$	$\Phi_l^{\gamma^2}$	$\Phi_l^*$
101	2.12	0.16	0.47	2.16	0.17	1.53
211	12.4	1.64	3.26	12.7	1.68	7.95
307	43.3	4.82	8.34	44.0	4.93	19.3
401	109	10.9	17.9	111	11.1	38.0
503	215	23.3	34.0	219	23.8	66.4
601	390	40.5	55.8	395	41.4	110
701	695	69.1	90.2	703	70.3	158
809	1130	105	134	1150	107	222
907	1590	158	194	1600	160	306
1009	2300	223	267	2320	225	403
2003	23900	2400	2590	24100	2440	3210
3001	106000	9250	9650	107000	9360	11200
4001	283000	25100	25900	287000	25400	28600
5003	647000	57000	58300	653000	60200	65700
10009	7180000	681000	687000	7320000	688000	713000

TABLE 3. Computations of  $\Phi_l$  and  $\Phi_l^{\gamma^2}$  modulo  $m$ .

Columns  $\Phi_l^*$  list the total time to obtain  $\Phi_l$  by computing  $\Phi_l^{\gamma^2}$  and applying (22).

## ACKNOWLEDGMENTS

We thank David Harvey for the `zn_poly` library, and Andreas Enge for providing timings for his evaluation/interpolation algorithm. We also thank Igor Shparlinski for his helpful comments on an early draft of this paper.

## APPENDIX

**Lemma 8.1.** *Let  $c$  be a real number greater than  $c_0 = \log_2 e \approx 1.44$ . Let  $\pi_c(x)$  count the integers  $n \in [3, x]$  for which  $\omega(n) \geq c \log \log n$ . Then  $\pi_c(x) = O(x(\log x)^{1-c/c_0})$ .*

*Proof.* Let  $d(n)$  count the divisors of  $n$ . From [34, Thm. 320] we have

$$\sum_{n \leq x} 2^{\omega(n)} \leq \sum_{n \leq x} d(n) = x \log x + O(x).$$

At most  $O(x(\log x)^{1-c/c_0})$  terms on the LHS have  $n \geq \sqrt{x}$  and  $\omega(n) \geq c \log \log x$ . Applying  $\sqrt{x} = O(x(\log x)^{1-c/c_0})$  and  $\frac{\log \log \sqrt{x}}{\log \log x} = 1 + o(1)$  yields the lemma.  $\square$

## REFERENCES

1. Eric Bach, *Explicit bounds for primality testing and related problems*, Mathematics of Computation **55** (1990), no. 191, 355–380.
2. Juliana Belding, Reinier Bröker, Andreas Enge, and Kristin Lauter, *Computing Hilbert class polynomials*, Algorithmic Number Theory Symposium–ANTS VIII (A. J. van der Poorten and A. Stein, eds.), Lecture Notes in Computer Science, vol. 5011, Springer, 2008, pp. 282–295.
3. Elwyn R. Berlekamp, *Factoring polynomials over large finite fields*, Mathematics of Computation **24** (1970), no. 111, 713–735.
4. Daniel J. Bernstein, *Modular exponentiation via the explicit Chinese Remainder Theorem*, Mathematics of Computation **76** (2007), 443–454.

5. Ingrid Biehl and Johannes Buchmann, *An analysis of the reduction algorithms for binary quadratic forms*, Voronoi's Impact on Modern Science (P. Engel and H. 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 Andrew V. Sutherland, *Computing the endomorphism ring of an ordinary elliptic curve over a finite field*, Journal of Number Theory (2009), to appear, <http://arxiv.org/abs/0902.4670>.
7. Ian Blake, Gadiel Seroussi, and Nigel Smart, *Elliptic curves in cryptography*, London Mathematical Society Lecture Note Series, vol. 265, Cambridge University Press, 1999.
8. Ian F. Blake, János A. Csirik, Michael Rubinstein, and Gadiel Seroussi, *On the computation of modular polynomials for elliptic curves*, Tech. report, Hewlett-Packard Laboratories, 1999, <http://www.math.uwaterloo.ca/~mrubinst/publications/publications.html>.
9. Alin Bostan, Bruno Salvy, François Morain, and Éric Schost, *Fast algorithms for computing isogenies between elliptic curves*, Mathematics of Computation **77** (2008), 1755–1778.
10. Reinier Bröker, *Constructing elliptic curves of prescribed order*, PhD thesis, Universiteit Leiden, 2006.
11. ———, *A  $p$ -adic algorithm to compute the Hilbert class polynomial*, Mathematics of Computation **77** (2008), 2417–2435.
12. ———,  *$p$ -adic class invariants*, LMS Journal of Computation and Mathematics (2010), to appear.
13. Reinier Bröker and Andrew V. Sutherland, *An explicit height bound for the classical modular polynomial*, Ramanujan Journal **22** (2010), 293–313.
14. Johannes Buchmann and Ulrich Vollmer, *Binary quadratic forms: an algorithmic approach*, Algorithms and Computations in Mathematics, vol. 20, Springer, 2007.
15. J.J. Cannon and W. Bosma (Eds.), *Handbook of Magma functions*, 2.15 ed., 2008, available at <http://magma.maths.usyd.edu.au/magma/htmlhelp/MAGMA.htm>.
16. Guilhem Castagnos and Fabien Laguillaumie, *On the security of cryptosystems with quadratic decryption: the nicest cryptanalysis*, Advances in Cryptology: EUROCRYPT 2009 (A. Joux, ed.), Lecture Notes in Computer Science, vol. 5479, Springer, 2009, pp. 260–277.
17. Denis Charles and Kristin Lauter, *Computing modular polynomials*, LMS Journal of Computation and Mathematics **8** (2005), 195–204.
18. Henri Cohen, *Advanced topics in computational number theory*, Springer, 2000.
19. Henri Cohen and Gerhard Frey et al., *Handbook of elliptic and hyperelliptic curve cryptography*, Chapman and Hall, 2006.
20. Paula Cohen, *On the coefficients of the transformation polynomials for the elliptic modular function*, Math. Proc. of the Cambridge Philosophical Society **95** (1984), 389–402.
21. David A. Cox, *Primes of the form  $x^2 + ny^2$ : Fermat, class field theory, and complex multiplication*, John Wiley and Sons, 1989.
22. Noam D. Elkies, *Elliptic and modular curves over finite fields and related computational issues*, Computational Perspectives on Number Theory (D. A. Buell and J. T. Teitelbaum, eds.), Studies in Advanced Mathematics, vol. 7, AMS, 1998, pp. 21–76.
23. Andreas Enge, *The complexity of class polynomial computation via floating point approximations*, Mathematics of Computation **78** (2009), 1089–1107.
24. ———, *Computing modular polynomials in quasi-linear time*, Mathematics of Computation **78** (2009), 1809–1824.
25. Andreas Enge and François Morain, *Generalized Weber functions I*, 2009, <http://arxiv.org/abs/0905.3250>.
26. Andreas Enge and Reinhard Schertz, *Constructing elliptic curves over finite fields using double eta-quotients*, Journal de Théorie des Nombres de Bordeaux **16** (2004), no. 3, 555–568.
27. ———, *Modular curves of composite level*, Acta Arithmetica **118** (2005), no. 2, 129–141.
28. Andreas Enge and Andrew V. Sutherland, *Class invariants for the CRT method*, Algorithmic Number Theory Symposium–ANTS IX (G. Hanrot, F. Morain, and E. Thomé, eds.), Lecture Notes in Computer Science, vol. 6197, Springer-Verlag, 2010, pp. 142–156.
29. Free Software Foundation, *GNU compiler collection*, January 2010, version 4.4.3, available at <http://gcc.gnu.org/>.
30. Mireille Fouquet and François Morain, *Isogeny volcanoes and the SEA algorithm*, Algorithmic Number Theory Symposium–ANTS V (C. Fieker and D. R. Kohel, eds.), Lecture Notes in Computer Science, vol. 2369, Springer, 2002, pp. 276–291.

31. Steven D. Galbraith, Florian Hess, and Nigel P. Smart, *Extending the GHS Weil descent attack*, Advances in Cryptology—EUROCRYPT 2002, Lecture Notes in Computer Science, vol. 2332, Springer, 2002, pp. 29–44.
32. Alice Gee and Peter Stevenhagen, *Generating class fields with Shimura reciprocity*, Algorithmic Number Theory Symposium—ANTS III, Lecture Notes in Computer Science, vol. 1423, Springer, 1998, pp. 442–453.
33. Torbjörn Granlund et al., *GNU multiple precision arithmetic library*, September 2010, version 5.0.1, available at <http://gmplib.org/>.
34. Godfrey H. Hardy and Edward M. Wright, *An introduction to the theory of numbers*, fifth ed., Oxford Science Publications, 1979.
35. David Harvey, *zn\_poly: a library for polynomial arithmetic*, 2008, version 0.9, [http://cims.nyu.edu/~harvey/zn\\_poly](http://cims.nyu.edu/~harvey/zn_poly).
36. ———, *Faster polynomial multiplication via multipoint Kronecker substitution*, Journal of Symbolic Computation **44** (2009), no. 10, 1502–1510.
37. Oskar Herrmann, *Über die Berechnung der Fourierkoeffizienten der Funktion  $j(\tau)$* , J. Reine Agnew. Math. **274/275** (1975), 187–195.
38. Derek F. Holt, Bettina Eick, and Eamonn A. O'Brien, *Handbook of computational group theory*, CRC Press, 2005.
39. Hideji Ito, *Computation of the modular equation*, Proc. Japan Acad. Ser. A **71** (1995), 48–50.
40. Moshe Jarden, *Transfer principles for finite and  $p$ -adic fields*, Nieuw Archief voor Wiskunde **3** (1980), no. 28, 139–158, <http://www.tau.ac.il/~jarden/Articles/paper27.pdf>.
41. Erich Kalfoten and Noriko Yui, *On the modular equation of order 11*, Proceedings of the 1984 MACSYMA Users Conference, 1984, pp. 472–485.
42. David Kohel, *Endomorphism rings of elliptic curves over finite fields*, PhD thesis, University of California at Berkeley, 1996.
43. J. C. Lagarias and A. M. Odlyzko, *Effective versions of the Chebotarev density theorem*, Algebraic number fields:  $L$ -functions and Galois properties (Proc. Sympos., Univ. Durham, Duram, 1975), Academic Press, 1977, pp. 409–464.
44. Serge Lang, *Elliptic functions*, second ed., Springer-Verlag, 1987.
45. Frank Lehmann, Markus Maurer, Volker Müller, and Victor Shoup, *Counting the number of points on elliptic curves over finite fields of characteristic greater than three*, Algorithmic Number Theory Symposium—ANTS I (L. M. Adleman and M.-D. Huang, eds.), Lecture Notes in Computer Science, vol. 877, 1994, pp. 60–70.
46. 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.
47. François Morain, *Calcul du nombre de points sur une courbe elliptique dans un corps fini: aspects algorithmiques*, Journal de Théorie des Nombres de Bordeaux **7** (1995), no. 1, 111–138.
48. Volker Müller, *Ein Algorithmus zur Bestimmung der Punktzahl elliptischer Kurven über endlichen Körpern der Charakteristik größer drei*, PhD thesis, Universität des Saarlandes, 1995.
49. Jürgen Neukirch, *Algebraic number theory*, Springer, 1999.
50. Arnold Schönhage and Volker Strassen, *Schnelle Multiplikation großer Zahlen*, Computing **7** (1971), 281–292.
51. René Schoof, *Counting points on elliptic curves over finite fields*, Journal de Théorie des Nombres de Bordeaux **7** (1995), 219–254.
52. William Stein and David Joyner, *SAGE: System for Algebra and Geometry Experimentation*, Communications in Computer Algebra (SIGSAM Bulletin) (2005), 61–64.
53. Peter Stevenhagen, *The arithmetic of number rings*, Algorithmic Number Theory: Lattices, Number Fields, and Cryptography (J.P. Buhler and P. Stevenhagen, eds.), Mathematical Sciences Research Institute Publications, vol. 44, Cambridge University Press, 2008.
54. Andrew V. Sutherland, *Computing Hilbert class polynomials with the Chinese Remainder Theorem*, Mathematics of Computation **80** (2011), 501–538.
55. Jacques Vélou, *Isogénies entre courbes elliptiques*, Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences, Séries A et B **273** (1971), 238–241.
56. Joachim von zur Gathen and Jürgen Gerhard, *Modern computer algebra*, second ed., Cambridge University Press, 2003.
57. Lawrence C. Washington, *Elliptic curves: Number theory and cryptography*, second ed., CRC Press, 2008.

58. Heinrich Weber, *Lehrbuch der algebra*, third ed., vol. III, Chelsea, 1961.

BROWN UNIVERSITY, BOX 1917, 151 THAYER STREET, PROVIDENCE, RHODE ISLAND 02192  
*E-mail address:* `reinier@math.brown.edu`

MICROSOFT RESEARCH, ONE MICROSOFT WAY, REDMOND, WASHINGTON 98052  
*E-mail address:* `klauter@microsoft.com`

MASSACHUSETTS INSTITUTE OF TECHNOLOGY, CAMBRIDGE, MASSACHUSETTS 02139  
*E-mail address:* `drew@math.mit.edu`