COMPUTING ENDOMORPHISM RINGS OF JACOBIANS OF GENUS 2 CURVES OVER FINITE FIELDS

DAVID FREEMAN AND KRISTIN LAUTER

Abstract. We present probabilistic algorithms which, given a genus 2 curve C defined over a finite field and a quartic CM field K, determine whether the endomorphism ring of the Jacobian J of C is the full ring of integers in K. In particular, we present algorithms for computing the field of definition of, and the action of Frobenius on, the subgroups $J[\ell^d]$ for prime powers ℓ^d. We use these algorithms to create the first implementation of Eisenträger and Lauter's algorithm for computing Igusa class polynomials via the Chinese Remainder Theorem [EL], and we demonstrate the algorithm for a few small examples. We observe that in practice the running time of the CRT algorithm is dominated not by the endomorphism ring computation but rather by the need to compute p^3 curves for many small primes p.

1. Introduction

Many public-key cryptographic protocols are based on the difficulty of the discrete logarithm problem in groups of points on elliptic curves and Jacobians of hyperelliptic curves. For such protocols one needs to work in a subgroup of large prime order of the Jacobian of the curve, so it is useful to be able to construct curves over finite fields whose Jacobians have a specified number of points.

The problem of constructing elliptic curves over finite fields with a given number of points has been studied extensively. Current solutions rely on computing the j-invariant via the construction of the Hilbert class polynomial for a quadratic imaginary field. There are three different approaches to computing the Hilbert class polynomial: a complex-analytic algorithm [AM], [Eng]; a Chinese Remainder Theorem algorithm [CNST], [ALV]; and a p-adic algorithm [CH], [Brö]. The best running time for these algorithms is $\tilde{O}(|d|)$, where d is the discriminant of the quadratic imaginary field [Eng], [Brö].

Analogous methods exist for constructing genus 2 curves with a given number of points on their Jacobians. In this case, the solutions rely on computing the curves' Igusa invariants via the computation of Igusa class polynomials for quartic CM fields. Again there are three different approaches: a complex-analytic algorithm [Spa], [vW], [W], [CL]; a Chinese Remainder Theorem algorithm [EL]; and a p-adic algorithm [GHKRW]. These algorithms are less extensively developed than their elliptic curve analogues, and to date there is no running time analysis for any of them.

In this paper we study the implementation of Eisenträger and Lauter's Chinese Remainder Theorem algorithm [EL]. The algorithm takes as input a primitive quartic CM field K, i.e. a purely imaginary quadratic extension of a real quadratic

Date: May 30, 2007.

field with no proper imaginary quadratic subfields, and produces the Igusa class polynomials of K. The basic outline of the algorithm is as follows:

(1) Define S to be a set of primes with certain splitting behavior in the field K and its reflex field K^*.
(2) For each prime p in S:
   (a) For each triple $(i_1, i_2, i_3) \in \mathbb{F}_p^3$ of Igusa invariants, construct a genus 2 curve C over $\mathbb{F}_p$ corresponding to that triple.
   (b) Check the isogeny class of each curve. For each curve in the desired isogeny class, compute the endomorphism ring of the Jacobian of the curve and keep only those curves for which the endomorphism ring is the full ring of integers $\mathcal{O}_K$.
   (c) Construct the Igusa class polynomials mod p from the triples collected in Step 2b.
(3) Use the Chinese Remainder Theorem or the Explicit CRT [Ber] to construct the Igusa polynomials either with rational coefficients or modulo a prime of cryptographic size.

One advantage of the CRT algorithm over other algorithms for computing Igusa class polynomials is that the CRT algorithm does not require that the real quadratic subfield have class number one.

Our contribution is to provide an efficient probabilistic algorithm for computing endomorphism rings of Jacobians of genus 2 curves over small prime fields. Using this algorithm to compute endomorphism rings, we have implemented a probabilistic version of the full Eisenträger-Lauter CRT algorithm (Algorithm 7.1) in MAGMA and used it to compute Igusa class polynomials for several fields K with small discriminant.

It was previously believed that computing endomorphism rings would be the bottleneck in the genus 2 CRT algorithm. Our results are surprising in the sense that we find that the time taken to compute the endomorphism rings with our probabilistic algorithms is negligible compared with the time needed to compute p^3 genus 2 curves via Mestre's algorithm for each small prime p. For example, for $K = \mathbb{Q}(i\sqrt{13 + 2\sqrt{13}})$ and $p = 157$, the largest prime for which endomorphism rings are computed for this K, our (unoptimized) MAGMA program takes about 52 minutes to loop through 157^3 curves and find 243 curves in the specified isogeny class. Our probabilistic algorithm (also implemented in MAGMA) applied to these 243 curves then takes 16.5 *seconds* to find the single curve whose Jacobian has endomorphism ring equal to $\mathcal{O}_K$.

The algorithm works as follows. Let C be a genus 2 curve over a finite field $\mathbb{F}_p$, and let J be its Jacobian; we assume J is ordinary. Let K be a primitive quartic CM field, which we assume is given via an embedding in $\mathbb{C}$. The first test is whether $\mathrm{End}(J)$, the endomorphism ring of J, is an order in $\mathcal{O}_K$. This computation is outlined in [EL, Section 5] and described in more detail in Section 2 below. If $\mathrm{End}(J)$ is an order in $\mathcal{O}_K$, we compute a set of possible elements $\pi \in \mathcal{O}_K$ that could represent the Frobenius endomorphism of J. If π represents the Frobenius endomorphism, then its complex conjugate $\overline{\pi}$ represents the Verschiebung endomorphism.

We next determine a set $\{\alpha_i\}$ of elements of $\mathcal{O}_K$ such that $\mathbb{Z}[\pi, \overline{\pi}, \{\alpha_i\}] = \mathcal{O}_K$. It follows that $\mathrm{End}(J) = \mathcal{O}_K$ if and only if each α_i is an endomorphism of J. We show in Section 3 that we can take each α_i to have one of two forms: either

$\alpha_i = \frac{\pi^k - 1}{\ell}$ for some positive integer k and prime ℓ, or $\alpha_i = \frac{h_i(\pi)}{\ell^d}$ for some cubic polynomial h_i with integer coefficients and some prime power ℓ^d. In Section 4 we show how to determine whether an element of the first form is an endomorphism; this is equivalent to determining the field of definition of the ℓ-torsion points of J. In Section 5 we show how to determine whether an element of the second form is an endomorphism; this is equivalent to computing the action of Frobenius on a basis of $J[\ell^d]$. The main results are Algorithms 4.3 and 5.1, two very efficient probabilistic algorithms which check fields of definition and compute the action of Frobenius, respectively. The running times of these algorithms depend primarily on the sizes of the fields over which the points of $J[\ell^d]$ are defined. Section 6 provides upper bounds for these sizes in terms of the prime ℓ and the size of the base field p.

A detailed statement of the Eisenträger-Lauter CRT algorithm, incorporating the algorithms of Sections 2, 4, and 5, appears in Section 7. Section 8 describes various ways in which we have modified our MAGMA implementation to improve the algorithm's performance. Finally, in Section 9 we give examples of our algorithm run on several small quartic CM fields.

Notation and assumptions. Throughout this paper, a *curve* will refer to a smooth, projective, absolutely irreducible algebraic curve C. The Jacobian of C, denoted $\mathrm{Jac}(C)$, is an abelian variety of dimension g, where g is the genus of C. We assume throughout that p is a prime, and that $\mathrm{Jac}(C)$ is an ordinary abelian variety modulo p.

A number field K is a *CM field* if it is a totally imaginary quadratic extension of a totally real field. We denote by K^* the reflex field of K, and by K_0 the real quadratic subfield of K. A CM field is *primitive* if it has no proper CM subfields. We will assume unless otherwise noted that K is a primitive quartic CM field not isomorphic to $\mathbb{Q}(\zeta_5)$. This implies that K is either Galois cyclic or non-Galois. If K is Galois cyclic, then $K^* = K$; if K is non-Galois, then K^* is another primitive quartic CM field [Shi, p. 64]. A curve C has *CM by* K if the endomorphism ring of $\mathrm{Jac}(C)$ is isomorphic to an order in $\mathcal{O}_K$, the ring of integers of the CM field K.

Acknowledgments. This research was conducted during the first author's internship at Microsoft Research, Redmond, during the summer of 2006. The first author thanks Microsoft for its hospitality and Denis Charles, Jean-Marc Couveignes, and Edward Schaefer for many helpful discussions. The second author thanks Pierrick Gaudry for helpful correspondence and pointers to his code. Both authors thank Reinier Bröker, David Kohel, and Christophe Ritzenthaler for their feedback on previous versions of this paper.

2. Computing zeta functions and the Frobenius element

To determine whether the Jacobian J of a given genus 2 curve C has endomorphism ring equal to $\mathcal{O}_K$, the first step is to determine whether the endomorphism ring is even an order in $\mathcal{O}_K$. This is accomplished by computing the characteristic polynomial of Frobenius, to see if the Frobenius element corresponds to an algebraic integer of K. This in turn is equivalent to determining the zeta function of C, which can be computed by finding the number of points on the curve and its Jacobian, $n = \#C(\mathbb{F}_p)$ and $m = \#J(\mathbb{F}_p)$. For a given field K there are several possibilities for the pairs (n, m), as described in [EL, Prop. 4].

In this section we give an explicit algorithm that determines whether $\mathrm{End}(J)$ is an order in $\mathcal{O}_K$ and if so, gives a set $S \subset \mathcal{O}_K$ of possibilities for the Frobenius endomorphism of J. The main point is to find the possible Frobenius elements by finding generators of certain principal ideals (Step 2) with absolute value equal to $\sqrt{p}$ (Step 4a).

Algorithm 2.1. Let K be a primitive quartic CM field and K^* the reflex of K. The following algorithm takes as input the field K, a prime p that splits completely in K and splits completely into principal ideals in K^*, and a curve C defined over the finite field $\mathbb{F}_p$. The algorithm returns `true` or `false` according to whether $\mathrm{End}(J)$ is an order in $\mathcal{O}_K$, where $J = \mathrm{Jac}(C)$. If the answer is `true`, the algorithm also outputs a set $S \subset \mathcal{O}_K$ that consists of the $\mathrm{Aut}(K/\mathbb{Q})$-orbit of the Frobenius endomorphism of J.

(1) Compute the decomposition $p = \mathfrak{p}_1\mathfrak{p}_2\mathfrak{p}_3\mathfrak{p}_4$ in $\mathcal{O}_K$, using e.g. [Coh, Alg. 6.2.9]. Renumber so that $\mathfrak{p}_2 = \overline{\mathfrak{p}_1}$ and $\mathfrak{p}_3 = \overline{\mathfrak{p}_4}$.
(2) Compute generators α_1 and α_2 for the principal ideals $\mathfrak{p}_1\mathfrak{p}_3$ and $\mathfrak{p}_2\mathfrak{p}_3$, respectively, using e.g. [Coh, Alg. 6.5.10].
(3) Compute a fundamental unit u of K_0 with $|u| > 1$, using e.g. [Coh, Alg. 5.7.1].
(4) For $i \leftarrow 1, 2$, do the following:
    (a) If $|\alpha_i| < \sqrt{p}$, set $\alpha_i \leftarrow \alpha_i u$ until $|\alpha_i| = \sqrt{p}$. If $|\alpha_i| > \sqrt{p}$, set $\alpha_i \leftarrow \alpha_i u^{-1}$ until $|\alpha_i| = \sqrt{p}$.
    (b) Compute the characteristic polynomial $h_i(x)$ of α_i, using e.g. [Coh, Prop. 4.3.4].
    (c) If K is Galois and $h_1(x) = h_2(-x)$, set $\alpha_2 \leftarrow -\alpha_2$ and $h_2(x) \leftarrow h_2(-x)$.
    (d) Set $(n_{i,+1}, m_{i,+1}) \leftarrow (p + 1 - \frac{h_i'(0)}{p}, h_i(1))$. Set $(n_{i,-1}, m_{i,-1}) \leftarrow (p + 1 + \frac{h_i'(0)}{p}, h_i(-1))$.
(5) Determine whether the Frobenius endomorphism of J has characteristic polynomial equal to $h_i(\pm x)$ for some i:
    (a) Choose a random point $P \in J(\mathbb{F}_p)$ and compute $Q_{j,\tau} = [m_{i,\tau}]P$ for $i \in \{1, 2\}$, $\tau \in \{\pm 1\}$. If none of $Q_{i,\tau}$ is the identity, return `false`. Otherwise, optionally repeat with another random point P.
    (b) If J passes a certain fixed number of trials of Step 5a, compute $\#C(\mathbb{F}_p)$. If $\#C(\mathbb{F}_p) \neq n_{i,\tau}$ for all $i \in \{1, 2\}$, $\tau \in \{\pm 1\}$, return `false`.
    (c) If $\#C(\mathbb{F}_p) = n_{i,\tau}$, compute $\#J(\mathbb{F}_p)$, using e.g. Baby Step Giant Step [Coh, Alg 5.4.1]. If $\#J \neq m_{i,\tau}$ for the same i, τ, return `false`.
(6) If K is Galois, output $S = \{\tau\alpha_1, \tau\overline{\alpha_1}, \tau\alpha_2, \tau\overline{\alpha_2}\}$. If K is not Galois, output $S = \{\tau\alpha_i, \tau\overline{\alpha_i}\}$, using the i determined in Step 5c.
(7) Return `true`.

Proof. The proof of [EL, Prop. 4] shows that the ideals $\mathfrak{p}_1\mathfrak{p}_3$ and $\mathfrak{p}_2\mathfrak{p}_3$ are principal and the Frobenius endomorphism of J corresponds to a generator of one of these ideals or their complex conjugates. Furthermore, this generator must have complex absolute value $\sqrt{p}$. The generators determined in Step 2 are unique up to unit multiple, so Step 4a ensures that the absolute values are $\sqrt{p}$, thus making each α_i unique up to complex conjugation and sign.

If the Frobenius element corresponds to α_i or $\overline{\alpha_i}$, then $h_i(x)$ is the characteristic polynomial of Frobenius, so we can determine this case by checking whether

$\#C(\mathbb{F}_p) = n_{i,+1}$ and $\#J(\mathbb{F}_p) = m_{i,+1}$. Similarly, if the Frobenius element corresponds to $-\alpha_i$ or $-\overline{\alpha_i}$, then $h_i(-x)$ is the characteristic polynomial of Frobenius, so we can determine this case by checking whether $\#C(\mathbb{F}_p) = n_{i,-1}$ and $\#J(\mathbb{F}_p) = m_{i,-1}$.

If K is Galois (with Galois group C_4), then the ideal (α_2) is equal to $(\alpha_1)^\sigma$ for some σ generating the Galois group. Since complex absolute value squared is the same as the norm from K to its real quadratic subfield K_0, $|\alpha_1| = \sqrt{p}$ implies that $|\alpha_1^\sigma| = \sqrt{p}$. Since α_1^σ and α_2 both generate (α_2) and have absolute value $\sqrt{p}$, we deduce that $\alpha_1^\sigma = \pm\alpha_2$. Step 4c ensures that this sign is positive, so α_1 and α_2 have the same characteristic polynomial $h_i(x)$, and thus the Frobenius element could be any of the elements output by Step 6. Since $\mathrm{Aut}(K/\mathbb{Q})$ is generated by σ and σ^2 is complex conjugation, we have output the $\mathrm{Aut}(K/\mathbb{Q})$-orbit of the Frobenius element.

If K is not Galois, then the Frobenius element must be either α_i or $\overline{\alpha_i}$. Since $\mathrm{Aut}(K/\mathbb{Q})$ in this case consists of only the identity and complex conjugation, Step 6 outputs the $\mathrm{Aut}(K/\mathbb{Q})$-orbit of the Frobenius element. □

3. Constructing a generating set for $\mathcal{O}_K$

Given the Jacobian J of a genus 2 curve over $\mathbb{F}_p$ and a primitive quartic CM field K, Algorithm 2.1 allows us to determine whether there is some $\pi \in \mathcal{O}_K$ that represents the Frobenius endomorphism of J. Since the complex conjugate $\overline{\pi}$ represents the Verschiebung endomorphism, if Algorithm 2.1 outputs `true` then we have

$$\mathbb{Z}[\pi, \overline{\pi}] \subseteq \mathrm{End}(J) \subseteq \mathcal{O}_K. \tag{3.1}$$

In this section, we assume we are given a $J/\mathbb{F}_p$ and a π such that (3.1) holds, and we wish to determine whether $\mathrm{End}(J) = \mathcal{O}_K$.

Let $\mathcal{B}$ be a $\mathbb{Z}$-module basis for $\mathcal{O}_K$, and consider the collection of elements $\{\alpha \in \mathcal{B} \setminus \mathbb{Z}\}$. Since this collection generates $\mathcal{O}_K$ over $\mathbb{Z}[\pi, \overline{\pi}]$, it suffices to determine whether or not each element of the collection is an endomorphism of J. Assuming K satisfies some mild hypotheses, Eisenträger and Lauter give one example of a basis $\mathcal{B}$ that suffices to determine the endomorphism ring [EL, Lemma 6]. However, the method given in [EL] lacks an efficient procedure for testing whether a given $\alpha \in \mathcal{B}$ is an endomorphism of J.

In this section, we derive from an arbitrary basis $\mathcal{B}$ a set of generators for $\mathcal{O}_K$ over $\mathbb{Z}[\pi, \overline{\pi}]$ that are convenient in the sense that there is an efficient probabilistic algorithm (Algorithm 4.3 or Algorithm 5.1) for determining whether an element of the set is an endomorphism of J. Our findings are summarized in Proposition 3.8.

We begin by observing that since $K = \mathbb{Q}(\pi)$, any $\alpha \in \mathcal{O}_K$ can be expressed as a polynomial $f \in \mathbb{Q}[\pi]$. Since π satisfies a polynomial of degree 4 (the characteristic polynomial of Frobenius), f can be taken to have degree 3. We may thus write

$$\alpha = \frac{a_0 + a_1\pi + a_2\pi^2 + a_3\pi^3}{n} \tag{3.2}$$

for some integers a_0, a_1, a_2, a_3, n. We assume that a_0, a_1, a_2, a_3 have no common factor with n, so that n is the smallest integer such that $n\alpha \in \mathbb{Z}[\pi]$.

Remark 3.1. The LLL lattice reduction algorithm [LLL], as implemented by the MAGMA command `LinearRelation`, finds an expression of the form (3.2) for any

$\alpha \in \mathcal{O}_K$. Given as input the sequence $[1, \pi, \pi^2, \pi^3, -\alpha]$, the algorithm outputs a sequence $[a_0, a_1, a_2, a_3, n]$ satisfying the relation (3.2).

The following lemma shows that each $\alpha \in \mathcal{B} \setminus \mathbb{Z}$ can be replaced with a collection of elements that generate the same ring, each with a power of a single prime in the denominator of the expression (3.2).

Lemma 3.2. *Let $A \subset B$ be commutative rings with 1, with $[B : A]$ finite. Suppose $\alpha \in B$, and let n be the smallest integer such that $n\alpha \in A$. Suppose n factors into primes as $\ell_1^{d_1} \cdots \ell_r^{d_r}$. Then*

$$A[\alpha] = A\Big[\tfrac{n}{\ell_1^{d_1}}\alpha, \ldots, \tfrac{n}{\ell_r^{d_r}}\alpha\Big].$$

Proof. Clearly the ring on the right is contained in the ring on the left, so we must show that α is contained in the ring on the right. It suffices to show that there are integers c_i such that

$$c_1 \frac{n}{\ell_1^{d_1}} + \cdots + c_r \frac{n}{\ell_r^{d_r}} = 1, \tag{3.3}$$

for then we can multiply this identity by α to get our result. We use the extended Euclidean algorithm and induct on r, the number of distinct primes dividing n. If $r = 1$ the result is trivial, for in this case $n/\ell_1^{d_1} = 1$. Now suppose (3.3) holds for any n that is divisible by r distinct primes. If n' is divisible by $r+1$ distinct primes, we can write $n' = n\ell_{r+1}^{d_{r+1}}$ for some n divisible by r distinct primes. Since ℓ_{r+1} is relatively prime to n, we can use the extended Euclidean algorithm to write $a\ell_{r+1}^{d_{r+1}} + bn = 1$ for some integers a, b. We can then multiply the first term by the left-hand side of (3.3) (which is equal to 1) to get

$$ac_1 \frac{n\ell_{r+1}^{d_{r+1}}}{\ell_1^{d_1}} + \cdots + ac_r \frac{n\ell_{r+1}^{d_{r+1}}}{\ell_r^{d_r}} + bn = ac_1 \frac{n'}{\ell_1^{d_1}} + \cdots + ac_r \frac{n'}{\ell_r^{d_r}} + b\frac{n'}{\ell_{r+1}^{d_{r+1}}} = 1.$$

This is an equation of the form (3.3) for n', which completes the proof. □

The next lemma shows that only primes dividing the index $[\mathcal{O}_K : \mathbb{Z}[\pi]]$ appear in the denominators.

Lemma 3.3. *Let α be an element of $\mathcal{O}_K$, and suppose n is the smallest integer such that $n\alpha \in \mathbb{Z}[\pi]$. Then n divides the index $[\mathcal{O}_K : \mathbb{Z}[\pi]]$.*

Proof. Let $N = [\mathcal{O}_K : \mathbb{Z}[\pi]]$. By definition, N is the size of the abelian group $\mathcal{O}_K/\mathbb{Z}[\pi]$. Thus we can write any $\alpha \in \mathcal{O}_K$ as $\alpha = a + b$ with $b \in \mathbb{Z}[\pi]$ and $N \cdot a \in \mathbb{Z}[\pi]$. This shows that $\mathcal{O}_K$ is contained in $\frac{1}{N}\mathbb{Z}[\pi]$. We may thus write $\alpha = f(\pi)/N$ for a unique polynomial f with integer coefficients and degree at most 3. Furthermore, since $n\alpha$ is the smallest multiple of α in $\mathbb{Z}[\pi]$, we may write $\alpha = g(\pi)/n$ for a unique polynomial g with integer coefficients and degree at most 3, such that n has no factor in common with all the coefficients of g. We thus have $n \cdot f(\pi) = N \cdot g(\pi)$. If we let d be the gcd of the coefficients of f and e be the gcd of the coefficients of g, then we have $n \cdot d = N \cdot e$.

Let ℓ be a prime dividing e. Since $\gcd(n, e) = 1$, ℓ must divide d, so we can cancel ℓ from both sides and get $n \cdot d' = N \cdot e'$ with $e' < e$. Proceeding in this manner until $e' = 1$, we conclude that n divides N. □

We now know that each $\alpha \in \mathcal{B} \setminus \mathbb{Z}$ can be replaced with a collection of elements $\{\frac{n}{\ell_i^{d_i}}\alpha\}$, and the only ℓ_i appearing are divisors of the index the index $[\mathcal{O}_K : \mathbb{Z}[\pi]]$. The following lemma and corollary show that for any ℓ which divides $[\mathcal{O}_K : \mathbb{Z}[\pi]]$ exactly (i.e. $\ell \mid [\mathcal{O}_K : \mathbb{Z}[\pi]]$ and $\ell^2 \nmid [\mathcal{O}_K : \mathbb{Z}[\pi]]$), the element $\frac{n}{\ell}\alpha$ can be replaced by an element of the form $\frac{\pi^k-1}{\ell}$. This replacement is useful since by [EL, Fact 10], determining whether an element of the form $\frac{\pi^k-1}{\ell}$ is an endomorphism is equivalent to testing the field of definition of the ℓ-torsion.

Lemma 3.4. *Let $A \subset B \subset C$ be abelian groups, with $[C : A]$ finite. Let ℓ be a prime, and suppose ℓ divides $[C : A]$ and ℓ^2 does not divide $[C : A]$. Suppose there is some $\beta \in B$ such that $\beta \notin A$ and $\ell\beta \in A$. Then for any $\alpha \in C$ such that $\ell\alpha \in A$, $\alpha \in B$.*

Proof. The hypotheses on $[C : A]$ imply that the ℓ-primary part of C/A (denoted $(C/A)_\ell$) is isomorphic to $\mathbb{Z}/\ell\mathbb{Z}$, so $(B/A)_\ell$ is either trivial or $\mathbb{Z}/\ell\mathbb{Z}$. The conditions on β imply that β has order ℓ in B/A, so $(B/A)_\ell \cong \mathbb{Z}/\ell\mathbb{Z} \cong (C/A)_\ell$, with the isomorphism induced by the inclusion map $B \hookrightarrow C$. Since α is in the ℓ-primary part of C/A, α must also be in the ℓ-primary part of B/A, so $\alpha \in B$. □

Corollary 3.5. *Suppose ℓ divides $[\mathcal{O}_K : \mathbb{Z}[\pi]]$ exactly and $\beta = \frac{\pi^k-1}{\ell} \notin \mathbb{Z}[\pi]$. Then $\frac{\pi^k-1}{\ell}$ is an endomorphism of J if and only if any $\alpha \in \mathcal{O}_K \setminus \mathbb{Z}[\pi]$ with $\ell\alpha \in \mathbb{Z}[\pi]$ is also an endomorphism.*

Proof. The result follows directly from Lemma 3.4, with $A = \mathbb{Z}[\pi]$, $B = \mathrm{End}(J)$, and $C = \mathcal{O}_K$. □

Furthermore, if $p \nmid [\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$, then any element α_i with denominator $\ell_i = p$ may be ignored due to the following corollary.

Corollary 3.6. *Suppose $p \nmid [\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$. Then for any $\alpha \in \mathcal{O}_K$ such that $p\alpha \in \mathbb{Z}[\pi]$, $\alpha \in \mathbb{Z}[\pi, \overline{\pi}]$.*

Proof. Since π is the Frobenius element, it satisfies a characteristic polynomial of the form

$$\pi^4 + s_1\pi^3 + s_2\pi^2 + s_1p\pi + p^2 = 0. \tag{3.4}$$

Using $\pi\overline{\pi} = p$ and dividing this equation by π gives

$$\pi^3 + s_1\pi^2 + s_2\pi + s_1p + p\overline{\pi} = 0. \tag{3.5}$$

From this equation we see that $p\overline{\pi} \in \mathbb{Z}[\pi]$, so either $[\mathbb{Z}[\pi, \overline{\pi}] : \mathbb{Z}[\pi]] = p$ or $\overline{\pi} \in \mathbb{Z}[\pi]$. If $\overline{\pi} \in \mathbb{Z}[\pi]$ then p divides the coefficients of the terms on the left hand side of (3.5), which it does not, so we deduce that $\overline{\pi} \notin \mathbb{Z}[\pi]$ and $[\mathbb{Z}[\pi, \overline{\pi}] : \mathbb{Z}[\pi]] = p$. The hypothesis $p \nmid [\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$ thus implies that p divides $[\mathcal{O}_K : \mathbb{Z}[\pi]]$ exactly, so we may apply Lemma 3.4 with $\ell = p$, $A = \mathbb{Z}[\pi]$, $B = \mathbb{Z}[\pi, \overline{\pi}]$, $C = \mathcal{O}_K$, and $\beta = \overline{\pi}$. □

Thus any α satisfying the conditions of the corollary is automatically an endomorphism. We now show that the condition $p \nmid [\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$ is automatically satisfied for all primes p except possibly 2 and 3.

Proposition 3.7. *Suppose $p > 3$ and that $\pi \in \mathcal{O}_K$ corresponds to the Frobenius endomorphism of an ordinary abelian surface A over $\mathbb{F}_p$. Then $p \nmid [\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$.*

Proof. Let $\Delta(R)$ denote the discriminant of a $\mathbb{Z}$-module R. Christophe Ritzenthaler pointed out that this proposition follows from [How, Proposition 9.4], which shows that

$$\Delta(\mathbb{Z}[\pi,\overline{\pi}]) = \pm \mathrm{Norm}_{K/\mathbb{Q}}(\pi-\overline{\pi})\,\Delta(\mathbb{Z}[\pi+\overline{\pi}]).$$

Alternatively, it is shown in [LPP, Proposition 7.4] that any prime that divides the index $[\mathcal{O}_K : \mathbb{Z}[\pi,\overline{\pi}]]$ must divide either $[\mathcal{O}_{K_0} : \mathbb{Z}[\pi+\overline{\pi}]]$ or $\frac{\Delta(\mathcal{O}_{K_0}[\pi])}{\Delta(\mathcal{O}_K)}$, and, using [How, Theorem 1.3], that the second quantity is prime to p if the abelian surface is ordinary. The same proposition also shows that $\Delta(\mathbb{Z}[\pi+\overline{\pi}]) < 16p$, and since

$$\frac{\Delta(\mathbb{Z}[\pi+\overline{\pi}])}{\Delta(\mathcal{O}_{K_0})} = [\mathcal{O}_{K_0} : \mathbb{Z}[\pi+\overline{\pi}]]^2,$$

we conclude that if p divides $[\mathcal{O}_K : \mathbb{Z}[\pi,\overline{\pi}]]$ then p^2 divides $[\mathcal{O}_{K_0} : \mathbb{Z}[\pi+\overline{\pi}]]^2$, and thus

$$p^2 \leq \frac{\Delta(\mathbb{Z}[\pi+\overline{\pi}])}{\Delta(\mathcal{O}_{K_0})} < \frac{16p}{5}$$

(since a real quadratic field has discriminant at least 5), which implies $p \leq 3$. □

The following proposition summarizes the results of this section.

Proposition 3.8. *Suppose $\{\alpha_i\}$ generates $\mathcal{O}_K$ as a $\mathbb{Z}$-algebra. Let n_i be the smallest integer such that $n_i\alpha_i \in \mathbb{Z}[\pi]$, and write the prime factorization of n_i as $n_i = \prod_j \ell_{ij}^{d_{ij}}$. For each (i,j) with $\ell_{ij} \neq p$, let k_{ij} be an integer such that $\pi^{k_{ij}} - 1 \in \ell_{ij}\mathcal{O}_K$. Suppose $p > 3$. Then the following set generates $\mathcal{O}_K$ over $\mathbb{Z}[\pi,\overline{\pi}]$:*

$$\left\{ \frac{n_i}{\ell_{ij}^{d_{ij}}}\alpha_i : \ell_{ij}^2 \mid [\mathcal{O}_K : \mathbb{Z}[\pi]] \right\} \cup \left\{ \frac{\pi^{k_{ij}}-1}{\ell_{ij}} : \ell_{ij}^2 \nmid [\mathcal{O}_K : \mathbb{Z}[\pi]], \ell_{ij} \neq p \right\}.$$

Remark 3.9. Proposition 3.8 shows that if $p > 3$ and the index $[\mathcal{O}_K : \mathbb{Z}[\pi,\overline{\pi}]]$ is square-free, then $\mathcal{O}_K$ can be generated over $\mathbb{Z}[\pi,\overline{\pi}]$ by a collection of elements of the form $\frac{\pi^k-1}{\ell}$. This answers a question raised by Eisenträger and Lauter [EL, Remark 5].

In our application, $\pi \in \mathcal{O}_K$ is only determined up to an automorphism of K, but Proposition 3.8 can still be used to determine a generating set for $\mathcal{O}_K$.

Corollary 3.10. *Let $\mathcal{S} \subset \mathcal{O}_K$ be the set given in Proposition 3.8. Let σ be an element of $\mathrm{Aut}(K/\mathbb{Q})$. Then the set $\{\beta^\sigma : \beta \in \mathcal{S}\}$ generates $\mathcal{O}_K$ over $\mathbb{Z}[\pi^\sigma,\overline{\pi}^\sigma]$.*

Proof. By Proposition 3.8, the set $\{\pi,\overline{\pi}\} \cup \mathcal{S}$ generates $\mathcal{O}_K$ as a $\mathbb{Z}$-algebra. Since $\mathcal{O}_K$ is mapped to itself by $\mathrm{Aut}(K/\mathbb{Q})$, the set $\{\pi^\sigma,\overline{\pi}^\sigma\} \cup \{\beta^\sigma : \beta \in \mathcal{S}\}$ also generates $\mathcal{O}_K$ as a $\mathbb{Z}$-algebra. The statement follows immediately. □

4. Determining fields of definition

In this section, we consider the problem of determining the field of definition of the n-torsion points of the Jacobian J of a genus 2 curve over $\mathbb{F}_p$. By [EL, Fact 10], the n-torsion points of J are defined over $\mathbb{F}_{p^k}$ if and only if $(\pi^k - 1)/n$ is an endomorphism of J, where π is the Frobenius endomorphism of J. Thus determining the field of definition of the ℓ-torsion points allows us to determine whether some of the elements given by Proposition 3.8 are endomorphisms.

Algorithm 4.1. The following algorithm takes as input a primitive quartic CM field K, an element $\pi \in \mathcal{O}_K$ with $\pi\overline{\pi} = p$, and an integer n with $\gcd(n, p) = 1$, and outputs the smallest integer k such that $\pi^k - 1 \in n\mathcal{O}_K$. If J is the Jacobian of a genus 2 curve over $\mathbb{F}_p$ with Frobenius π^σ for some $\sigma \in \mathrm{Aut}(K/\mathbb{Q})$ and $\mathrm{End}(J) = \mathcal{O}_K$, this integer k is such that the n-torsion points of J are defined over $\mathbb{F}_{p^k}$.

(1) Compute a $\mathbb{Z}$-basis $\mathcal{B} = (1, \delta, \gamma, \kappa)$ of $\mathcal{O}_K$, using [SW] or [Coh, Algorithm 6.1.8], and write $\pi = (a, b, c, d)$ in this basis. Set $k \leftarrow 1$.
(2) Let $\bar{\mathcal{B}}$ be the reduction of the elements of $\mathcal{B}$ modulo n. Let $(a_1, b_1, c_1, d_1) = (a, b, c, d) \pmod{n}$.
(3) Compute $\pi^k \equiv (a_k, b_k, c_k, d_k) \pmod{n}$ with respect to $\bar{\mathcal{B}}$.
(4) If $(a_k, b_k, c_k, d_k) \equiv (1, 0, 0, 0) \pmod{n}$, output k. Otherwise set $k \leftarrow k + 1$ and go to Step 3.

Proof. The set $\bar{\mathcal{B}}$ is a $\mathbb{Z}/n\mathbb{Z}$-basis of $\mathcal{O}_K/n\mathcal{O}_K$, so if $\pi^k \equiv (1, 0, 0, 0) \pmod{n}$, then $\pi^k - 1 \in n\mathcal{O}_K$ (since the first element of $\bar{\mathcal{B}}$ is 1). Since $n\mathcal{O}_K$ is mapped to itself by $\mathrm{Aut}(K/\mathbb{Q})$, we have $(\pi^\sigma)^k - 1 \in n\mathcal{O}_K$. If $\mathrm{End}(J) = \mathcal{O}_K$, then $\frac{(\pi^\sigma)^k - 1}{n} \in \mathcal{O}_K = \mathrm{End}(J)$, so by [EL, Fact 10], $J[n] \subset J(\mathbb{F}_{p^k})$. □

Remark 4.2. Since $J[n] = \bigoplus J[\ell^d]$ for prime powers ℓ^d dividing n, we may speed up Algorithm 4.1 by factoring n and computing $k(\ell^d)$ for each prime power factor ℓ^d; then $k(n) = \mathrm{lcm}(k(\ell^d))$. Furthermore, we will see in Propositions 6.2 and 6.3 below that for a fixed ℓ^d, the possible values of k are very limited. Thus we may speed up the algorithm even further by precomputing these possible values and testing each one, rather than increasing the value of k by 1 until the correct value is found.

Eisenträger and Lauter [EL] computed endomorphism rings in several examples by determining the group structure of $J(\mathbb{F}_{p^k})$ to decide whether $J[n] \subset J(\mathbb{F}_{p^k})$. This is an exponential-time algorithm that is efficient only for very small k. Eisenträger and Lauter also suggested that the algorithm of Gaudry-Harley [GH] could be used to determine the field of definition of the n-torsion points. One of the primary purposes of this article is to present an efficient probabilistic algorithm to test the field of definition of $J[n]$. Below we describe the various methods of testing the field of definition of the n-torsion of J. Since $J[n] = \bigoplus J[\ell^d]$ as ℓ^d ranges over maximal prime-power divisors of n, it suffices to consider each prime-power factor separately. We thus assume in what follows that $n = \ell^d$ is a prime power.

4.1. **The brute force method.** The simplest method of determining the field of definition of the n-torsion is to compute the abelian group structure of $J(\mathbb{F}_{p^k})$. The MAGMA syntax for this computation is straightforward, and the program returns a group structure of the form

$$J(\mathbb{F}_{p^k}) \cong \frac{\mathbb{Z}}{a_1\mathbb{Z}} \times \cdots \times \frac{\mathbb{Z}}{a_j\mathbb{Z}},$$

with $a_1 \mid \cdots \mid a_j$. The n-torsion of J is contained in $J(\mathbb{F}_{p^k})$ if and only if $j = 4$ and n divides a_1.

While this method is easy to implement, if k is too large it may take too long to compute the group structure (via Baby-Step/Giant-Step or similar algorithms), or even worse we may not even be able to factor $\#\,\mathrm{Jac}(C)(\mathbb{F}_{p^k})$. In practice, computing group structure in MAGMA seems to be feasible for group sizes up to roughly 2^{200},

which means p^k should be no more than roughly 2^{100}, and thus k will have to be very small. Thus the brute force method is very limited in scope; however, it has the advantage that in the small cases it can handle it runs fairly quickly and always outputs the right answer.

4.2. **The Gaudry-Harley-Schost method.** Gaudry and Harley [GH] define a Schoof-Pila-like algorithm for counting points on genus 2 curves. The curves input to this algorithm are assumed to have a degree 5 model over $\mathbb{F}_q$, so we can write elements of the Jacobian as pairs of affine points minus twice the Weierstrass point at infinity. An intermediate step in the algorithm is to construct a polynomial $R(x) \in \mathbb{F}_q[x]$ with the following property: if P_1 and P_2 are points on C such that $D = [P_1] + [P_2] - 2[\infty]$ is an n-torsion point of J, then the x-coordinates of P_1 and P_2 are roots of R. The field of definition of the x-coordinates is at most a degree-two extension of the field of definition of D. Thus in many cases the field of definition of the n-torsion points can be determined from the factorization of $R(x)$.

Gaudry has implemented the algorithm in MAGMA and NTL; the algorithm involves taking two resultants of pairs of two-variable polynomials of degree roughly n^2. The algorithm uses the clever trick of computing a two-variable resultant by computing many single-variable resultants and interpolating the result. The interpolation only works if the field of definition of J has at least $4n^2 - 8n + 4$ elements, so we must base extend J until the field of definition is large enough. Since $R(x)$ has coefficients in $\mathbb{F}_p$, this base extension has no effect on the result of the computation.

Gaudry and Harley's analysis of the algorithm gives a running time of $\tilde{O}(n^6)$ field multiplications if fast polynomial arithmetic is used, and $O(n^8)$ otherwise. Due to its large space requirements, the algorithm has only succeeded at handling inputs of size $n \leq 19$ [GS].

4.3. **A probabilistic method.** As usual, we let J be the Jacobian of a genus 2 curve over $\mathbb{F}_{p^k}$, and $\ell \neq p$ be a prime. Let H be the ℓ-primary part of $J(\mathbb{F}_{p^k})$. Then H has the structure

$$H = \frac{\mathbb{Z}}{\ell^{\alpha_1}\mathbb{Z}} \times \frac{\mathbb{Z}}{\ell^{\alpha_2}\mathbb{Z}} \times \frac{\mathbb{Z}}{\ell^{\alpha_3}\mathbb{Z}} \times \frac{\mathbb{Z}}{\ell^{\alpha_4}\mathbb{Z}},$$

with $\alpha_1 \leq \alpha_2 \leq \alpha_3 \leq \alpha_4$. Our test rests on the following observations:

- If the ℓ^d-torsion points of J are defined over $\mathbb{F}_{p^k}$, then $\alpha_1 \geq d$, and the number of ℓ^d-torsion points in H is ℓ^{4d}.
- If the ℓ^d-torsion points of J are not defined over $\mathbb{F}_{p^k}$, then $\alpha_1 < d$, and the number of ℓ^d-torsion points in H is at most ℓ^{4d-1}.

We thus make the following calculation: write $\#J(\mathbb{F}_{p^k}) = \ell^s m$ with $\ell \nmid m$. Choose a random point $P \in J$. Then $[m]P \in H$, and we test whether $[\ell^d m]P = O$ in J. If the ℓ^d-torsion points of J are defined over $\mathbb{F}_{p^k}$, then $[\ell^d m]P = O$ with probability $\rho = \ell^{4d-s}$, while if the ℓ^d-torsion points of J are not defined over $\mathbb{F}_{p^k}$ then $[\ell^d m]P = O$ with probability at most ρ/ℓ. If we perform the test enough times, we can determine which probability distribution we are observing and thus conclude, with a high degree of certainty, whether the ℓ^d-torsion points are defined over $\mathbb{F}_{p^k}$.

This method is very effective in practice, and can be implemented for large k: while computing the group structure of $J(\mathbb{F}_{p^k})$ for large k may be infeasible, it is

much easier to compute *points* on $J(\mathbb{F}_{p^k})$ and to do arithmetic on those points. We now give a formal description of the algorithm and determine its probability of success.

Algorithm 4.3. The following algorithm takes as input the Jacobian J of a genus 2 curve defined over a finite field $\mathbb{F}_q$, a prime power ℓ^d with $\gcd(\ell, q) = 1$, and a real number $\epsilon \in (0, 1)$. If $J[\ell^d] \subset J(\mathbb{F}_q)$, then the algorithm outputs `true` with probability at least $1 - \epsilon$. If $J[\ell^d] \not\subset J(\mathbb{F}_q)$, then the algorithm outputs `false` with probability at least $1 - \epsilon$.

(1) Compute $\#J(\mathbb{F}_q) = \ell^s m$, where $\ell \nmid m$. If $s < 4d$ output `false`.
(2) Set $\rho \leftarrow \ell^{4d-s}$, $N \leftarrow \lceil \frac{\sqrt{-2\log\epsilon}}{\rho}(\frac{2\ell}{\ell-1})\rceil$, $B \leftarrow \rho N\left(\frac{\ell+1}{2\ell}\right)$.
(3) Repeat N times:
    (a) Choose a random point $P_i \in J(\mathbb{F}_q)$.
    (b) Compute $Q_i \leftarrow [\ell^d m]P_i$
(4) If at least B of the Q_i are the identity element O of J, output `true`; otherwise output `false`.

Proof. As observed above, if $J[\ell^d] \subset J(\mathbb{F}_q)$, then $Q_i = O$ with probability ρ, while if $J[\ell^d] \not\subset J(\mathbb{F}_q)$, then $Q_i = O$ with probability at most ρ/ℓ. Thus all we have to do is compute enough Q_i to distinguish the two probability distributions. To figure out how many "enough" is, we use the Chernoff bound [Ros, Ch. 8, Prop. 5.3]. The version of the bound we use is as follows: If N weighted coins are flipped and μ is the expected number of heads, then for any $\delta \in (0, 1]$ we have

$$\begin{aligned} \Pr[\#\text{heads} < (1-\delta)\mu] &< e^{-\mu^2\delta^2/2} \\ \Pr[\#\text{heads} > (1+\delta)\mu] &< e^{-\mu^2\delta^2/2}. \end{aligned} \tag{4.1}$$

In our case we are given two different probability distributions for the coin flip and wish to tell them apart. If the ℓ^d-torsion points of J are defined over $\mathbb{F}_q$, then the probability that $Q_i = O$ is $\rho = \ell^{4d}/\ell^s$. Thus the expected number of Q_i equal to O is $\mu_1 = \rho N$. If the ℓ^d-torsion points are not defined over $\mathbb{F}_q$, then the expected number of Q_i equal to O is at most $\mu_2 = \rho N/\ell$. Thus if we set $B = \rho N(\frac{\ell+1}{2\ell})$ to be the midpoint of $[\mu_2, \mu_1]$, we will deduce that $J[\ell^d] \subset J(\mathbb{F}_q)$ if the number of Q_i equal to O is at least B, and $J[\ell^d] \not\subset J(\mathbb{F}_q)$ otherwise.

We thus wish to find an N such that this deduction is correct with probability at least $1 - \epsilon$, i.e. an N such that

$$\begin{aligned} \Pr[\#\{Q_i : Q_i = O\} < B] &< \epsilon \quad \text{if } J[\ell^d] \subset J(\mathbb{F}_q), \\ \Pr[\#\{Q_i : Q_i = O\} > B] &< \epsilon \quad \text{if } J[\ell^d] \not\subset J(\mathbb{F}_q). \end{aligned} \tag{4.2}$$

Substituting our choice of B into the Chernoff bound (4.1) gives

$$\begin{aligned} \Pr[\#\{Q_i : Q_i = O\} < B] &< e^{-2\mu_1^2(\frac{\ell-1}{4\ell})^2} \quad \text{if } J[\ell^d] \subset J(\mathbb{F}_q), \\ \Pr[\#\{Q_i : Q_i = O\} > B] &< e^{-2\mu_2^2(\frac{\ell-1}{4})^2} \quad \text{if } J[\ell^d] \not\subset J(\mathbb{F}_q). \end{aligned} \tag{4.3}$$

From these equations, we see that we wish to have $2\mu_1^2(\frac{\ell-1}{4\ell})^2 > -\log\epsilon$ and $2\mu_2^2(\frac{\ell-1}{4})^2 > -\log\epsilon$. The two left sides are equal since $\mu_2 = \mu_1/\ell$. We thus substitute $\mu_1 = \rho N$ into the relation $2\mu_1^2(\frac{\ell-1}{4\ell})^2 > -\log\epsilon$, and find that

$$N > \frac{\sqrt{-2\log\epsilon}}{\rho} \cdot \frac{2\ell}{\ell-1}.$$

Thus this value of N suffices to give the desired success probabilities. □

Remark 4.4. If $s = 4d$, then the algorithm can be simplified considerably. In this case, if $J[\ell^d] \subset J(\mathbb{F}_q)$ then the ℓ-primary part H of $J(\mathbb{F}_q)$ is isomorphic to $(\mathbb{Z}/\ell^d\mathbb{Z})^4$, and if not then it contains a point of order greater than ℓ^d. Thus if $J[\ell^d] \subset J(F)$ then Q_i will always be the identity, and the algorithm will always return `true`. On the other hand, if $J[\ell^d] \not\subset J(\mathbb{F}_q)$, we may abort the algorithm and return `false` as soon as we find a point $Q_i \neq O$, for in this case we have found a point in H of too large order, and thus the ℓ^d-torsion points are not defined over $\mathbb{F}_q$. If $J[\ell^d] \not\subset J(\mathbb{F}_q)$, then the probability that a random point in H has order $\leq \ell^d$ is at most $1/\ell$, so we must conduct at least $N = \lceil \frac{-\log \epsilon}{\log \ell} \rceil$ trials to ensure a success probability of at least $1 - \epsilon$. Thus in this case the method may require many fewer trials.

Remark 4.5. Note that while $\#J(\mathbb{F}_q)$ may be very large, in our application where J is defined over a small prime field it is easy to compute $\#J(\mathbb{F}_q)$ from the zeta function of the curve of which J is the Jacobian. Furthermore, while it is probably impossible to factor $\#J(\mathbb{F}_q)$ completely in a reasonable amount of time, it is easy to determine the highest power of ℓ that divides $\#J(\mathbb{F}_q)$.

Proposition 4.6. *Let J be the Jacobian of a genus 2 curve over $\mathbb{F}_p$. Assume that the zeta function of $J/\mathbb{F}_p$ is known, so that the cost to compute $\#J(\mathbb{F}_{p^k}) = \ell^s m$ is negligible. Then the expected number of operations in $\mathbb{F}_p$ necessary to execute Algorithm 4.3 on $J/\mathbb{F}_{p^k}$ (ignoring $\log\log p$ factors) is*

$$O(k^2 \log k (\log^2 p) \ell^{s-4d} (-\log \epsilon)^{1/2})$$

Proof. We must compare the cost of the two actions of Step 3, repeated N times. Choosing a random point on $J(\mathbb{F}_q)$ is equivalent to computing a constant number of square roots in $\mathbb{F}_q$, and taking a square root requires $O(\log q)$ field operations in $\mathbb{F}_q$ (see [GG, Algorithm 14.15 and Corollary 14.16]). The order of $J(\mathbb{F}_q)$ is roughly q^2, so multiplying a point on $J(\mathbb{F}_q)$ by an integer using a binary expansion takes $O(\log q)$ point additions on $J(\mathbb{F}_q)$. Each point addition takes a constant number of field operations in $\mathbb{F}_q$, so we see that the time of each trial is $O(\log q) = O(k \log p)$. If fast multiplication techniques are used, then the number of field operations in $\mathbb{F}_p$ needed to perform one field operation in $\mathbb{F}_q$ is $O(\log q \log\log q) = O(k \log k \log p)$ (ignoring $\log\log p$ factors), so each trial takes $O(k^2 \log k \log^2 p)$ field operations in $\mathbb{F}_p$. The number of trials is $O(\ell^{s-4d}\sqrt{-\log \epsilon})$, which gives a total of $O(k^2 \log k (\log^2 p) \ell^{s-4d} (-\log \epsilon)^{1/2})$ field operations in $\mathbb{F}_p$. □

5. Computing the action of Frobenius

As in the previous section, we consider a genus 2 curve C over $\mathbb{F}_p$ with Jacobian J, and assume that the endomorphism ring of J is an order in the ring of integers $\mathcal{O}_K$ of a primitive quartic CM field K. We let π represent the Frobenius endomorphism, and we look at elements $\alpha \in \mathcal{O}_K$ such that $\ell^d \alpha \in \mathbb{Z}[\pi]$ for some prime power ℓ^d. We wish to devise a test that, given such an α, determines whether α is an endomorphism of J.

Since π satisfies a quartic polynomial with integer coefficients, we can write α as

$$\alpha = \frac{a_0 + a_1\pi + a_2\pi^2 + a_3\pi^3}{\ell^d} \tag{5.1}$$

for some integers a_0, a_1, a_2, a_3. Expressing α in this form is useful because of the following fact proved by Eisenträger and Lauter [EL, Corollary 9]: α is an endomorphism if and only if $T = a_0 + a_1\pi + a_2\pi^2 + a_3\pi^3$ acts as zero on the ℓ^d-torsion. Thus we need a method for determining whether T acts as zero on the ℓ^d-torsion. Since T is a linear operator, it suffices to check whether $T(Q_i)$ is zero for each Q_i in some set whose points span the full ℓ^d-torsion. Below we describe three different ways to compute such a spanning set.

5.1. **The brute force method.** The most straightforward way to compute a spanning set for the ℓ^d-torsion is to use group structure algorithms to compute a basis of $J[\ell^d]$. This method was used in [EL] to compute the class polynomials in one example. The methods of Section 4 determine a k for which $J[\ell^d] \subset J(\mathbb{F}_{p^k})$. The computation of the group structure of $J(\mathbb{F}_{p^k})$ gives generators for the group; multiplying these generators by appropriate integers gives generators for the ℓ^d-torsion. It is then straightforward to compute the action of T on each generator g_i for $1 \le i \le 4$. If $T(g_i) = O$ for all i, then α is an endomorphism; otherwise α is not an endomorphism.

This method of computing a spanning set has the same drawback as the brute-force method of computing fields of definition: since the best algorithm for computing group structure runs in time exponential in $k \log p$, the method becomes prohibitively slow as k increases. Thus the method is only effective when ℓ^d is very small.

5.2. **A probabilistic method.** The method of Section 5.1 for computing generators of $J[\ell^d]$ becomes prohibitively slow as the field of definition of the ℓ^d-torsion points becomes large. However, we can get around this obstacle by randomly choosing many points Q_i of exact order ℓ^d, so that it is highly probable that the set $\{Q_i\}$ spans $J[\ell^d]$.

Recall that we wish to test whether the operator $T = a_0 + a_1\pi + a_2\pi^2 + a_3\pi^3$ acts as zero on the ℓ^d-torsion. To perform the test, we determine the field $\mathbb{F}_{p^k}$ over which we expect the ℓ^d-torsion to be defined. (See Section 4.) We pick a random point $P \in J(\mathbb{F}_{p^k})$ and multiply P by an appropriate integer to get a point Q whose order is a power of ℓ. If Q has order ℓ^d, we act on Q by the operator T and test whether we get the identity of J; otherwise we try again with a new P. (See Section 5.3 for another method of randomly choosing ℓ^d-torsion points.) We repeat the test until it is overwhelmingly likely that the points Q span the ℓ^d-torsion. If the set of Q spans the ℓ^d-torsion, then α is an endomorphism if and only if T acts as zero on all the Q.

Algorithm 5.1. The following algorithm takes as input the Jacobian J of a genus 2 curve over $\mathbb{F}_q$ with CM by K, a prime power ℓ^d with $\gcd(\ell, q) = 1$, the element $\pi \in \mathcal{O}_K$ corresponding to the Frobenius endomorphism of J, an element $\alpha \in \mathcal{O}_K$ such that $\ell^d\alpha \in \mathbb{Z}[\pi]$, and a real number $\epsilon > 0$. The algorithm outputs `true` or `false`.

Suppose $J[\ell^d] \subset J(\mathbb{F}_q)$. If α is an endomorphism of J, then the algorithm outputs `true`. If α is not an endomorphism of J, then the algorithm outputs `false` with probability at least $1 - \epsilon$.

(1) Compute a_0, a_1, a_2, a_3 such that α satisfies equation (5.1).

(2) Set N to be

$$N \leftarrow \begin{cases} \lceil \frac{1}{d-\log_\ell 2}(-\log_\ell \epsilon + 3d) \rceil & \text{if } \ell^d > 2 \\ \max\{\lceil -2\log_2 \epsilon \rceil + 6, 16\} & \text{if } \ell^d = 2. \end{cases}$$

(3) Compute $\#J(\mathbb{F}_q) = \ell^s m$, where $\ell \nmid m$.
(4) Set $i \leftarrow 1$.
(5) Choose a random point $P_i \in J(\mathbb{F}_q)$. Set $Q_i \leftarrow [m]P_i$. Repeat until $[\ell^d]Q_i = O$ and $[\ell^{d-1}]Q_i \neq O$.
(6) Compute

$$[a_0]Q_i + [a_1]\operatorname{Frob}_p(Q_i) + [a_2]\operatorname{Frob}_{p^2}(Q_i) + [a_3]\operatorname{Frob}_{p^3}(Q_i) \tag{5.2}$$

in $J(\mathbb{F}_q)$. If the result is nonzero output `false`.
(7) If $i < N$, set $i \leftarrow i + 1$ and go to Step 5.
(8) Output `true`.

Proof. By [EL, Corollary 9], α is an endomorphism of J if and only if the expression (5.2) is O for all ℓ^d-torsion points Q. Furthermore, it suffices to check the the expression only on a basis of the ℓ^d-torsion. Step 5 repeats until we find a point Q_i of exact order ℓ^d; the assumption $J[\ell^d] \subset J(\mathbb{F}_q)$ guarantees that we can find such a point. The algorithm computes a total of N such points Q_i. Thus if the set of Q_i span $J[\ell^d]$, then the algorithm will output `true` or `false` correctly, according to whether $\alpha \in \operatorname{End}(J)$. We must therefore compute a lower bound for the probability that the set of Q_i computed span $J[\ell^d]$.

To compute this bound, we will compute an upper bound for the probability that N points of exact order ℓ^d do not span $J[\ell^d]$. We will make repeated use of the following inequality, which can be proved easily with simple algebra: if ℓ, d, n, and m are positive integers with $\ell > 1$ and $n > m$, then

$$\frac{\ell^{md} - \ell^{m(d-1)}}{\ell^{nd} - \ell^{n(d-1)}} < \frac{1}{\ell^{(n-m)d}} \tag{5.3}$$

Next we observe that in any group of the form $(\mathbb{Z}/\ell^d\mathbb{Z})^r$, there are $\ell^{rd} - \ell^{r(d-1)}$ elements of exact order ℓ^d. The probability that a set of N elements does not span a 4-dimensional space is the sum of the probabilities that all the elements span a j-dimensional subspace, for $j = 1, 2, 3$. We consider each case:

- $j = 1$: All of the Q_i are in the space spanned by Q_1, and Q_1 can be any element. The probability of this happening is

$$\left(\frac{\ell^d - \ell^{d-1}}{\ell^{4d} - \ell^{4(d-1)}}\right)^{N-1} < \left(\frac{1}{\ell^{3d}}\right)^{N-1}.$$

- $j = 2$: Q_1 can be any element, one of the Q_i must be independent of Q_1, and the remaining $N-2$ elements must be in the same 2-dimensional subspace. There are $N-1$ ways to choose the second element, so the total probability is

$$(N-1)\left(1 - \frac{\ell^d - \ell^{d-1}}{\ell^{4d} - \ell^{4(d-1)}}\right)\left(\frac{\ell^{2d} - \ell^{2(d-1)}}{\ell^{4d} - \ell^{4(d-1)}}\right)^{N-2} < N\left(\frac{1}{\ell^{2d}}\right)^{N-2}.$$

- $j = 3$: Q_1 can be any element, and there must be two more linearly independent elements; there are $\binom{N-1}{2}$ ways of choosing these elements. The

remaining $N-3$ elements must all be in the same 3-dimensional subspace, so the total probability is

$$\frac{(N-1)(N-2)}{2}\left(1-\frac{\ell^d-\ell^{d-1}}{\ell^{4d}-\ell^{4(d-1)}}\right)\left(1-\frac{\ell^{2d}-\ell^{2(d-1)}}{\ell^{4d}-\ell^{4(d-1)}}\right)\left(\frac{\ell^{3d}-\ell^{3(d-1)}}{\ell^{4d}-\ell^{4(d-1)}}\right)^{N-3} < \frac{N^2}{2}\left(\frac{1}{\ell^d}\right)^{N-3}.$$

Summing these three cases, we see that the total probability that the Q_i do not span $J[\ell^d]$ is bounded above by

$$N^2\left(\frac{1}{\ell^d}\right)^{N-3}. \tag{5.4}$$

Since $2^N \geq N^2$ for $N \geq 4$, we have

$$N^2\left(\frac{1}{\ell^d}\right)^{N-3} \leq \ell^{-dN+3d+N\log_\ell 2}.$$

(Note that $N \geq 4$ must always hold if we want to have a spanning set of $J[\ell]$.) Setting this last expression less than ϵ and taking logs, we find

$$N \geq \frac{1}{d-\log_\ell 2}\left(-\log_\ell \epsilon + 3d\right). \tag{5.5}$$

Thus if the number of trials N is greater than or equal to the right hand side of (5.5), then the probability of success is at least $1-\epsilon$.

The right hand side of expression (5.5) is undefined if $\ell = 2$, $d = 1$, so we must make a different estimate. Since $2^{N/2} \geq N^2$ for $N \geq 16$, the estimate (5.4) bounds the probability of Q_i not spanning $J[\ell^d]$ by

$$\frac{N^2}{2^{N-3}} \leq \frac{1}{2^{N/2-3}}.$$

Setting the right hand side less than ϵ and taking logs gives

$$N \geq -2\log_2 \epsilon + 6. \tag{5.6}$$

Thus if the number of trials N is greater than or equal to the maximum of 16 and the right hand side of (5.6), then the probability of success is at least $1-\epsilon$. □

Corollary 5.2. *Let J, ℓ^d, α, and ϵ be as in Algorithm 5.1. Suppose $\pi \in \mathcal{O}_K$ is such that π^σ corresponds to the Frobenius endomorphism of J for some $\sigma \in \mathrm{Aut}(K/\mathbb{Q})$. Suppose $J[\ell^d] \subset J(\mathbb{F}_q)$, and suppose Algorithm 5.1 is run with inputs J, $\mathbb{F}_q$, π, α, ϵ. If α^σ is an endomorphism of J, then the algorithm outputs* `true`*. If α^σ is not an endomorphism of J, then the algorithm outputs* `false` *with probability at least $1-\epsilon$.*

Proof. If we write α in the form (5.1), then we have

$$\alpha^\sigma = \frac{a_0 + a_1\pi^\sigma + a_2(\pi^\sigma)^2 + a_3(\pi^\sigma)^3}{\ell^d}. \tag{5.7}$$

Step 6 of the algorithm determines whether the numerator of this expression acts as zero on ℓ^d-torsion points. By [EL, Corollary 9], this action is identically zero if and only if α^σ is an endomorphism of J. The statement now follows from the correctness of Algorithm 5.1. □

Remark 5.3. Since Q_i is an ℓ^d-torsion point in Step 6, we may speed up the computation of the expression (5.2) by replacing each a_j with a small representative of a_j modulo ℓ^d. We may also rewrite the expression (5.2) as

$$[a_0]Q_i + \mathrm{Frob}_p([a_1]Q_i + \mathrm{Frob}_p([a_2]Q_i + \mathrm{Frob}_p([a_3]Q_i)))$$

to reduce the number of Frob_p operations from 6 to 3.

Remark 5.4. Algorithm 5.1 assumes that the ℓ^d-torsion points of J are defined over $\mathbb{F}_q$, so with enough trials we are almost certain to get a spanning set of points Q_i. However, if the ℓ^d-torsion points are not defined over $\mathbb{F}_q$, then the points Q_i will span a proper subspace of $J[\ell^d]$. If α is an endomorphism then T will act as zero on all of the Q_i and Algorithm 5.1 will output `true`. However, if α is not an endomorphism then T may still act as zero on all of the Q_i (in which case it must have nonzero action on the ℓ^d-torsion points that are not defined over $\mathbb{F}_q$), and the algorithm will incorrectly output `true`. Thus to test whether α is an endomorphism, we must combine Algorithm 5.1 with a method of checking the field of definition of the ℓ^d-torsion points, via the probabilistic method of Algorithm 4.3 or one of the other methods.

Proposition 5.5. *Let J be the Jacobian of a genus 2 curve over $\mathbb{F}_p$. Assume that the zeta function of $J/\mathbb{F}_p$ is known, so that the cost to compute $\#J(\mathbb{F}_{p^k}) = \ell^s m$ is negligible. Then the expected number of operations in $\mathbb{F}_p$ necessary to execute Algorithm 5.1 on $J/\mathbb{F}_{p^k}$ (ignoring $\log\log p$ factors) is*

$$O(k^2 \log k (\log^2 p) \ell^{s-4d}(-\log \epsilon))$$

Proof. Let $q = p^k$. In the proof of Proposition 4.6, we computed that the cost of computing a random point on $J(\mathbb{F}_q)$ is $O(\log q)$ operations in $\mathbb{F}_q$, and the cost of a point multiplication on $J(\mathbb{F}_q)$ is $O(\log q)$ operations in $\mathbb{F}_q$. The chance that a random point in the ℓ-primary part of $J(\mathbb{F}_q)$ has exact order ℓ is $\frac{\ell^{4d} - \ell^{4d-4}}{\ell^s}$, so the expected number of random points necessary to find one point of exact order ℓ^d is $O(\ell^{s-4d})$. The cost of computing the Frobenius action is proportional to the cost of raising an element of $\mathbb{F}_q$ to the pth power, which is $O(\log p)$ $\mathbb{F}_q$-operations.

We conclude that the expected cost of a single trial with a random point is

$$O(\log q + \log q + \log p)\ell^{s-4d} M(q)$$

operations in $\mathbb{F}_p$, where $M(q)$ is the number of field operations in $\mathbb{F}_p$ needed to perform one field operation in $\mathbb{F}_q$. If fast multiplication techniques are used, then $M(q) = O(\log q \log\log q) = O(k \log k \log p)$ (ignoring $\log\log p$ factors), so each trial takes $O(k^2 \log k (\log^2 p)\ell^{s-4d})$ field operations in $\mathbb{F}_p$. The number of points of exact order ℓ^d computed is $O(-\log\epsilon)$. Putting this all together gives a total of $O(k^2 \log k (\log^2 p)\ell^{s-4d}(-\log\epsilon))$ field operations in $\mathbb{F}_p$. □

5.3. **The Couveignes method.** Recall that to test whether an element $\alpha \in \mathcal{O}_K$ of the form (5.1) is an endomorphism of J, we determine whether the operator $T = a_0 + a_1\pi + a_2\pi^2 + a_3\pi^3$ acts as zero on all elements of a set $\{Q_i\}$ that spans $J[\ell^d]$. Algorithm 5.1 computes the spanning set by choosing random points P_i in $J(\mathbb{F}_{p^k})$, multiplying by an appropriate m to get points Q_i in the ℓ-primary part of $J(\mathbb{F}_{p^k})$ (denoted $J(\mathbb{F}_{p^k})_\ell$), and keeping only those Q_i whose order is exactly ℓ^d. If $J(\mathbb{F}_{p^k})_\ell$ is much larger than $J[\ell]$, the orders of most of the Q_i will be too large, and it will take many trials to find the required number of points of order exactly

ℓ^d. To reduce the number of trials required, we would like to find a function from $J(\mathbb{F}_{p^k})_\ell$ to $J[\ell^d]$ that sends most of the Q_i to points of exact order ℓ^d.

One way to compute such a function is as follows: compute the order ℓ^{t_i} of each Q_i; if $t_i \geq d$ send $Q_i \mapsto [\ell^{t-d}]Q_i$, otherwise send $Q_i \mapsto O$. In most cases the image has order ℓ^d. However, since the multiplier ℓ^{t_i-d} will be different for each Q_i, this function does not define a group homomorphism, and thus the image of a set of points uniformly distributed in $J(\mathbb{F}_{p^k})_\ell$ will not be uniformly distributed in $J[\ell^d]$.

Couveignes [Cou] has described a map that has the properties we want and is a group homomorphism. The idea is the following: if $\pi^k - 1 \in \ell^d \operatorname{End}(J)$, then there is an endomorphism ϕ such that $\ell^d\phi = \pi^k - 1$. Since $\pi^k - 1$ acts as zero on $J(\mathbb{F}_{p^k})$, the image of ϕ on $J(\mathbb{F}_{p^k})$ must consist of ℓ^d-torsion points. Furthermore, the kernel of ϕ contains $\ell^d J(\mathbb{F}_{p^k})$, since $\phi(\ell^d P) = (\pi^k - 1)(P) = 0$ if P is defined over $\mathbb{F}_{p^k}$. Thus we have a map

$$\phi : J(\mathbb{F}_{p^k})/\ell^d J(\mathbb{F}_{p^k}) \to J[\ell^d].$$

Couveignes then uses the non-degeneracy of the Frey-Rück pairing (see [Sch]) to show that ϕ is a bijection. Thus for any Q_i not in $\ell J(\mathbb{F}_{p^k})$, $\phi(Q_i)$ has order exactly ℓ^d. Since ϕ is a surjective group homomorphism, the image of a set of points uniformly distributed in $J(\mathbb{F}_{p^k})$ will be uniformly distributed in $J[\ell^d]$. The chance that $Q_i \in \ell J(\mathbb{F}_{p^k})$ is $1/\ell^4$, so applying ϕ to the Q_i will very quickly give a spanning set of $J[\ell^d]$.

However, there is one important caveat: we may not be able to compute ϕ. The only endomorphisms we can compute are those involving the action of Frobenius and scalar multiplication; namely, endomorphisms in $\mathbb{Z}[\pi]$. Thus we need to take k to be the smallest integer such that $\pi^k - 1 \in \ell^d\mathbb{Z}[\pi]$. We can then use the characteristic polynomial of Frobenius to write $\phi = \frac{\pi^k-1}{\ell^d} = M(\pi)$, where M is a polynomial of degree 3. Furthermore, since we are applying ϕ only to points $Q_i \in J(\mathbb{F}_{p^k})_\ell$, we may reduce the coefficients of M modulo ℓ^s and get the same action on the Q_i.

We have implemented the map ϕ in Magma and tested it on the examples that appear in Section 9. In our examples, the smallest k for which $\pi^k - 1 \in \ell^d\mathbb{Z}[\pi]$ is usually equal to ℓk_0, where k_0 is the integer output by Algorithm 4.1. We found that the cost of choosing random points over a field of degree ℓ times as large far outweighs the benefit of having to reject fewer of the points Q_i, so this technique does not help to speed up Algorithm 5.1.

6. Bounding the field of definition of the ℓ^d-torsion points

The running times of Algorithms 4.3 and 5.1 depend primarily on the size of the field $\mathbb{F}_{p^k}$ over which the ℓ^d-torsion points of J are defined. In this section, we bound the size of k in terms of ℓ^d and p. We also show that to determine the field of definition of the ℓ^d-torsion points of J for $d > 1$, it suffices to determine the field of definition of the ℓ-torsion points of J. This result allows us to work over much smaller fields in Algorithm 4.3, thus saving us a great deal of computation.

By Lemma 3.3, the prime powers ℓ^d input to Algorithms 4.3 and 5.1 divide the index $\mathbb{Z}[\pi, \overline{\pi}]$. Thus a bound on this index gives a bound on the ℓ^d that appear.

Proposition 6.1. *Let K be a primitive quartic CM field with discriminant $\Delta = \Delta(\mathcal{O}_K)$. Suppose $\pi \in \mathcal{O}_K$ corresponds to the Frobenius endomorphism of the Jacobian of a genus 2 curve defined over $\mathbb{F}_p$. Then*

$$[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]] \leq \frac{16p^2}{\sqrt{\Delta}}.$$

Proof. We showed in the proof of Corollary 3.6 that $[\mathbb{Z}[\pi, \overline{\pi}] : \mathbb{Z}[\pi]] = p$. Combining this result with the formula

$$[\mathcal{O}_K : \mathbb{Z}[\pi]] = [\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]][\mathbb{Z}[\pi, \overline{\pi}] : \mathbb{Z}[\pi]],$$

we see that it suffices to show that $[\mathcal{O}_K : \mathbb{Z}[\pi]] \leq 16p^3/\sqrt{\Delta}$. (Note that $\Delta > 0$ by [How, Proposition 9.4].) Next, recall that

$$[\mathcal{O}_K : \mathbb{Z}[\pi]] = \sqrt{\frac{\Delta(\mathbb{Z}[\pi])}{\Delta(\mathcal{O}_K)}}.$$

It thus suffices to show that $\sqrt{\Delta(\mathbb{Z}[\pi])} \leq 16p^3$. By definition,

$$\sqrt{\Delta(\mathbb{Z}[\pi])} = \prod_{i<j} |\alpha_i - \alpha_j|, \tag{6.1}$$

where α_i are the possible embeddings of π into $\mathbb{C}$. Since π represents an action of Frobenius, all of the α_i lie on the circle $|z| = \sqrt{p}$. The product (6.1) takes its maximum value subject to this constraint when the α_i are equally spaced around the circle, which happens when the α_i are $\sqrt{p}$ times primitive eighth roots of unity. The maximum product is thus $p^3\sqrt{\Delta(\mathbb{Q}(\zeta_8))} = 16p^3$. □

Proposition 6.1 also follows directly from [LPP, Proposition 7.4], where it is proved in a different manner that $\sqrt{\Delta(\mathbb{Z}[\pi, \overline{\pi}])} \leq 16p^2$.

The next two propositions give tight bounds on the degree k of the extension field of $\mathbb{F}_p$ over which the ℓ^d-torsion points of J are defined. The first considers the case $d = 1$, and the second shows that as d increases, k grows by a factor of ℓ^{d-1}.

Proposition 6.2. *Let J be the Jacobian of a genus 2 curve over $\mathbb{F}_p$, and suppose that $\mathrm{End}(J)$ is isomorphic to the ring of integers $\mathcal{O}_K$ of the primitive quartic CM field K. Let $\ell \neq p$ be a prime number, and suppose $\mathbb{F}_{p^k}$ is the smallest field over which the points of $J[\ell]$ are defined. If ℓ is unramified in K, then k divides one of the following:*

- *$\ell - 1$, if ℓ splits completely in K;*
- *$\ell^2 - 1$, if ℓ splits into two or three prime ideals in K;*
- *$\ell^3 - \ell^2 + \ell - 1$, if ℓ is inert in K.*

If ℓ ramifies in K, then k divides one of the following:

- *$\ell^3 - \ell^2$, if there is a prime over ℓ of ramification degree 3, or if ℓ is totally ramified in K and $\ell \leq 3$;*
- *$\ell^2 - \ell$, in all other cases where ℓ factors into four prime ideals in K (counting multiplicities);*
- *$\ell^3 - \ell$, if ℓ factors into two or three prime ideals in K (counting multiplicities).*

Proof. Let $\pi \in \mathcal{O}_K$ correspond to the Frobenius endomorphism. By [EL, Fact 10], the ℓ-torsion points of J are defined over $\mathbb{F}_{p^k}$ if and only if $\pi^k - 1 \in \ell\mathcal{O}_K$. We observe that by the Chinese Remainder Theorem, this condition is satisfied if and only if $\pi^k \equiv 1 \pmod{\mathfrak{p}_i^{e_i}}$ for all primes $\mathfrak{p}_i \mid \ell\mathcal{O}_K$, where e_i is the ramification degree of $\mathfrak{p}_i$. Next, we note that the condition $\ell \neq p$ implies that $\pi \notin \mathfrak{p}_i$ for all i. To see why this is true, suppose the contrary: $\pi \in \mathfrak{p}_i$. Since $\pi\overline{\pi} = p$, we have $p \in \mathfrak{p}_i$, contradicting the fact that $\mathfrak{p}_i$ is a prime over $\ell \neq p$.

From these observations we deduce that k is the least common multiple of the multiplicative orders of π mod each $\mathfrak{p}_i^{e_i}$, and thus k must divide the least common multiple of

$$\#(\mathcal{O}_K/\mathfrak{p}_i^{e_i}\mathcal{O}_K)^\times = \ell^{f_i(e_i-1)}(\ell^{f_i} - 1),$$

where f_i is the inertia degree of $\mathfrak{p}_i$. We now consider the various possibilities for the splitting of ℓ in $\mathcal{O}_K$.

First, suppose ℓ is unramified, so $e_i = 1$ for all i.

- If ℓ splits completely, then the inertia degrees of all the $\mathfrak{p}_i$ are 1, so $k \mid \ell - 1$.
- If ℓ splits into two or three ideals, then at least one $\mathfrak{p}_i$ has $f_i = 2$ and all have $f_i \leq 2$, so $k \mid \ell^2 - 1$.
- If ℓ is inert, then there is a single $\mathfrak{p}_i$ with $f_i = 4$, and k divides $\ell^4 - 1$. We will return to this case below to get a better bound.

Now suppose ℓ ramifies; there are six possibilities for the splitting of ℓ in $\mathcal{O}_K$.

- If $\ell\mathcal{O}_K = \mathfrak{p}^3\mathfrak{q}$, then $\mathfrak{p}$ and $\mathfrak{q}$ have inertia degree 1, so k divides $\ell^2(\ell - 1)$.
- If $\ell\mathcal{O}_K = \mathfrak{p}^4$, then $\mathcal{O}_K/\mathfrak{p} \cong \mathbb{F}_\ell$, and thus we have $\pi^{\ell-1} = 1 + \tau$ for some $\tau \in \mathfrak{p}$. There are now two subcases:
  - If $\ell \geq 5$, then $(1 + \tau)^\ell \in 1 + \mathfrak{p}^4$, so $\pi^{\ell(\ell-1)} \equiv 1 \pmod{\mathfrak{p}^4}$. Thus k divides $\ell(\ell - 1)$.
  - If $\ell = 2$ or 3, then $(1 + \tau)^\ell \equiv 1 + \tau^\ell \pmod{\mathfrak{p}^4}$, so we must raise the expression to the ℓth power again to get rid of the τ^ℓ term. Thus $\pi^{\ell^2(\ell-1)} \equiv 1 \pmod{\mathfrak{p}^4}$, and k divides $\ell^2(\ell - 1)$.
- If $\ell\mathcal{O}_K = \mathfrak{p}^2\mathfrak{q}^2$ or $\mathfrak{p}^2\mathfrak{q}\mathfrak{r}$, then all of the primes in question have inertia degree 1, so k divides $\ell(\ell - 1)$.
- If $\ell\mathcal{O}_K = \mathfrak{p}^2\mathfrak{q}$, then $\mathfrak{p}$ has inertia degree 1 and $\mathfrak{q}$ has inertia degree 2, so k divides $\operatorname{lcm}(\ell(\ell - 1), \ell^2 - 1) = \ell(\ell^2 - 1)$.
- If $\ell\mathcal{O}_K = \mathfrak{p}^2$, then $\mathcal{O}_K/\mathfrak{p} \cong \mathbb{F}_{\ell^2}$, and thus we have $\pi^{\ell^2-1} = 1 + \tau$ for some $\tau \in \mathfrak{p}$. Then $(1 + \tau)^\ell \in 1 + \mathfrak{p}^2$, so $\pi^{\ell(\ell^2-1)} \equiv 1 \pmod{\mathfrak{p}^2}$. Thus k divides $\ell(\ell^2 - 1)$.

Thus far we have used only the fact that π is an algebraic integer, and we have not used the property that it represents the action of Frobenius. To get a better bound in the case where ℓ is inert in K, we recall that since π is the Frobenius endomorphism, we have $\pi\overline{\pi} = p$, and $K = \mathbb{Q}(\pi)$. Since ℓ is inert, reduction modulo ℓ gives an injective group homomorphism

$$\phi\colon \operatorname{Aut}\left({}^{K}/_{\mathbb{Q}}\right) \to \operatorname{Aut}\left({}^{(\mathcal{O}_K/\ell\mathcal{O}_K)}/_{(\mathbb{Z}/\ell\mathbb{Z})}\right).$$

Furthermore, the target group is isomorphic to $\operatorname{Gal}(\mathbb{F}_{\ell^4}/\mathbb{F}_\ell)$. This group is cyclic of order 4 and is generated by the ℓth-power Frobenius automorphism. Since complex conjugation has order 2 in $\operatorname{Aut}(K/\mathbb{Q})$, its image under ϕ must be the map $\alpha \mapsto \alpha^{\ell^2}$. Thus $\overline{\pi} \equiv \pi^{\ell^2} \pmod{\ell}$, and $\pi^{\ell^2+1} \equiv p \pmod{\ell}$. Since p must reduce to an element of $\mathbb{F}_\ell^\times$, p has order dividing $\ell - 1$, so π must have order dividing $(\ell^2 + 1)(\ell - 1)$. $\square$

The following proposition shows that in the cases we need for our application, the field of definition of the ℓ^d-torsion points is determined completely by the field of definition of the ℓ-torsion points.

Proposition 6.3. *Let A be an ordinary abelian variety defined over a finite field F, and let ℓ be a prime number not equal to the characteristic of F. Let d be a positive integer, and let F' be the extension field of F of degree ℓ^{d-1}. If the ℓ-torsion points of A are defined over F, then the ℓ^d-torsion points of A are defined over F'. If $\mathrm{End}(A)$ is integrally closed, then the converse also holds.*

Proof. Let $R = \mathrm{End}(A)$, and let $\pi \in R$ be the Frobenius endomorphism of F. By [EL, Fact 10], for any positive integers t and k, the ℓ^t-torsion points of A are defined over the degree-k extension of F if and only if $\frac{\pi^k - 1}{\ell^t} \in R$, i.e. $\pi^k \equiv 1 \pmod{\ell^t R}$. To prove the proposition, it suffices to show that

$$\pi \equiv 1 \pmod{\ell R} \quad \Leftrightarrow \quad \pi^{\ell^{d-1}} \equiv 1 \pmod{\ell^d R},$$

with ($\Leftarrow$) holding when R is integrally closed.

First suppose that $\pi^k \equiv 1 \pmod{\ell^t R}$, with $t \geq 1$. Then we can write $\pi^k = 1 + \ell^t y$ for some $y \in R$. Then

$$\pi^{k\ell} = 1 + \ell(\ell^t y) + \binom{\ell}{2}(\ell^t y)^2 + \cdots + (\ell^t y)^\ell,$$

so $\pi^{k\ell} \equiv 1 \pmod{\ell^{t+1} R}$. We conclude that if the points of $A[\ell^t]$ are defined over the degree-k extension of F, then the points of $A[\ell^{t+1}]$ are defined over the degree-$k\ell$ extension of F. Thus if $A[\ell] \subset A(F)$, then by induction $A[\ell^d] \subset A(F')$.

Now suppose that $\pi^{k\ell} \equiv 1 \pmod{\ell^t R}$, with $t \geq 2$. Since A is ordinary, R is an order in a number ring. Thus if R is integrally closed then it is a Dedekind domain, and we may write $\ell R = \prod \mathfrak{p}_i^{e_i}$ uniquely for prime ideals $\mathfrak{p}_i \subset R$. By the Chinese Remainder Theorem, $\pi^k \equiv 1 \pmod{\ell^t R}$ if and only if $\pi^k \equiv 1 \pmod{\mathfrak{p}_i^{e_i t}}$ for each i, so we may consider the problem locally at each $\mathfrak{p}_i$. Localizing and completing the ring R at the prime $\mathfrak{p}_i$ gives a complete local ring R_v with maximal ideal $\mathfrak{p}_i$ and valuation v satisfying $v(\ell) = e_i$.

By hypothesis, we may write $\pi^{k\ell} = 1 + y$ for some $y \in \mathfrak{p}_i^{e_i t}$. We can define the ℓth-root function on R_v to be

$$(1+y)^{1/\ell} = \exp\left(\frac{1}{\ell}\log(1+y)\right). \tag{6.2}$$

By [Neu, Proposition II.5.5], if $y \in \mathfrak{p}_i^{e_i t}$ then $\log(1+y) \in \mathfrak{p}_i^{e_i t}$. Since $v(\ell) = e_i$, we have $v(\frac{1}{\ell}\log(1+y)) \geq e_i(t-1)$, so by the same Proposition $(1+y)^{1/\ell}$ converges and is in $1 + \mathfrak{p}_i^{e_i(t-1)}$ whenever $(t-1)(\ell-1) > 1$. Thus if $(t-1)(\ell-1) > 1$ then $\pi^k \equiv 1 \pmod{\mathfrak{p}_i^{e_i(t-1)}}$. We conclude that if $t > 2$ or $\ell > 2$ and the points of $A[\ell^t]$ are defined over the degree-$k\ell$ extension of F, then the points of $A[\ell^{t-1}]$ are defined over the degree-k extension of F. If $A[\ell^d] \subset A(F')$, then by descending induction $A[\ell] \subset A(F)$ if ℓ is odd, and $A[4] \subset A(F_2)$ if $\ell = 2$, where F_2 is the quadratic extension of F.

It remains to show that if $A[4] \subset A(F_2)$, then $A[2] \subset A(F)$. This is equivalent to showing that if $\pi^2 - 1 \in 4R$ then $\pi - 1 \in 2R$. We prove the contrapositive: suppose $\pi - 1 \notin 2R$. Then there is some prime $\mathfrak{p}$ over 2 such that $v_\mathfrak{p}(\pi - 1) < v_\mathfrak{p}(2)$. Since $\pi + 1 = (\pi - 1) + 2$ and $v_\mathfrak{p}(\pi - 1) < v_\mathfrak{p}(2)$, we must also have $v_\mathfrak{p}(\pi + 1) < v_\mathfrak{p}(2)$.

Multiplying the two expressions gives $v_{\mathfrak{p}}(\pi^2-1) < v_{\mathfrak{p}}(4)$, so π^2-1 cannot be contained in $4R$. We conclude that $\pi^2-1 \in 4R$ implies $\pi-1 \in 2R$. □

Corollary 6.4. *Let J be the Jacobian of a genus 2 curve over $\mathbb{F}_p$, and suppose that $\mathrm{End}(J)$ is isomorphic to the ring of integers $\mathcal{O}_K$ of the primitive quartic CM field K. Let ℓ^d be a prime power with $\ell \neq p$, and suppose $\mathbb{F}_{p^k}$ is the smallest field over which the points of $J[\ell^d]$ are defined. Then $k < 3p^6$.*

Proof. By Proposition 6.2, the points of $J[\ell]$ are defined over a field F of degree less than ℓ^3 over $\mathbb{F}_p$. By Proposition 6.3, the points of $J[\ell^d]$ are defined over a field L of degree ℓ^{d-1} over F. Since degrees of extensions multiply, we get

$$k = [L:\mathbb{F}_p] < \ell^{d+2} \leq \ell^{3d}.$$

By Proposition 6.1, $\ell^d \leq \frac{16}{\sqrt{\Delta}}p^2$, where Δ is the discriminant of the quartic CM field K. Lemma 6.5 below shows that any primitive quartic CM field has $\Delta \geq 125$, so $\ell^d \leq \frac{16}{\sqrt{125}}p^2$. Since $k < \ell^{3d}$, we conclude that $k < 3p^6$. □

Lemma 6.5. *Suppose K is a primitive quartic CM field. Then $\Delta(K) \geq 125$.*

Proof. Since $\Delta(\mathbb{Q}(\zeta_5)) = 125$, it suffices to show that no smaller discriminant can occur. The fact that $\Delta(K) > 0$ follows from [How, Proposition 9.4]. Now suppose $\Delta(K) < 125$. Since $\Delta(K_0)^2 \mid \Delta(K)$, we must have $K_0 = \mathbb{Q}(\sqrt{2})$ or $\mathbb{Q}(\sqrt{5})$, as these are the only two real quadratic fields with discriminant less than 12. Since $\mathbb{Q}(\sqrt{2})$ has class number 1, by [Neu, Proposition VI.6.9], $\mathbb{Q}(\sqrt{2})$ has no unramified quadratic extensions, so $\Delta(K)$ is strictly greater than $\Delta(K_0)^2$. Thus if $K_0 = \mathbb{Q}(\sqrt{2})$ then $\Delta(K) \geq 128$.

We deduce that $K_0 = \mathbb{Q}(\sqrt{5})$ and K must be of the form $\mathbb{Q}(i\sqrt{a+b\sqrt{5}})$, with a, b, and a^2-5b^2 positive integers. Since K is primitive, a^2-5b^2 is not a square in $\mathbb{Q}$ and its square-free part divides $\Delta(K)/\Delta(K_0)^2$. It thus suffices to show that the square-free part of a^2-5b^2 is at least 5; this follows from the fact that 2 and 3 are inert in $\mathbb{Q}(\sqrt{5})$, so there are no integer solutions to $a^2-5b^2 = 2$ or 3. □

7. Computing Igusa class polynomials

This section combines the results of all of the previous sections into a full-fledged probabilistic version of Eisenträger and Lauter's CRT algorithm to compute Igusa class polynomials for primitive quartic CM fields [EL, Theorem 1].

Algorithm 7.1. The following algorithm takes as input a primitive quartic CM field K, three integers $\lambda_1, \lambda_2, \lambda_3$ which are multiples of the denominators of the three Igusa class polynomials, and a real number $\epsilon > 0$, and outputs three polynomials $H_1, H_2, H_3 \in \mathbb{Q}[x]$. With high probability, the polynomials $H_i(x)$ output by the algorithm are the Igusa class polynomials for K.

(1) (Initialization.)
   (a) Let D be the degree of the Igusa class polynomials for K, computed via class number algorithms, e.g. [Coh, Algorithm 6.5.9].
   (b) Compute an integral basis $\mathcal{B}$ for $\mathcal{O}_K$, using e.g. [Coh, Algorithm 6.1.8].
   (c) Set $p \leftarrow 3$, $B \leftarrow 1$, $H_1, H_2, H_3 \leftarrow 0$, $F_1, F_2, F_3 \leftarrow 0$.

(2) Set $p \leftarrow$ `NextPrime`(p) until p splits completely in K and p splits into principal ideals in K^* (the reflex field of K).

(3) (Finding the curves.) Set $T_1, T_2, T_3 \leftarrow \{\}$. For each $(i_1, i_2, i_3) \in \mathbb{F}_p^3$, do the following:
   (a) Compute a curve $C/\mathbb{F}_p$ with Igusa invariants (i_1, i_2, i_3), using the algorithms of Mestre [Mes] and Cardona-Quer [CQ].
   (b) Run Algorithm 2.1 with inputs K, p, C.
      (i) If the algorithm outputs `false`, go to the next triple (i_1, i_2, i_3).
      (ii) If the algorithm outputs `true`, let π be one of the possible Frobenius elements it outputs.
   (c) For each prime ℓ dividing $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$, do the following:
      (i) Run Algorithm 4.1 with inputs K, ℓ, π. Let the output be k.
      (ii) Run Algorithm 4.3 with inputs $\mathrm{Jac}(C)$, $\mathbb{F}_{p^k}$, ℓ, and ϵ. If the output is `false`, go to the next triple (i_1, i_2, i_3).
      (iii) If ℓ^2 divides $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$, then for each $\alpha \in \mathcal{B} \setminus \mathbb{Z}$ written in the form (3.2) with denominator n, do the following:
         (A) Let d be the largest integer such that $\ell^d \mid n$. If $d = 0$, go to the next α.
         (B) Set $k' \leftarrow k\ell^{d-1}$.
         (C) Run Algorithm 5.1 with inputs $\mathrm{Jac}(C), \mathbb{F}_{p^{k'}}, \ell^d, \pi, \frac{n}{\ell^d}\alpha, \epsilon$.
         (D) If Algorithm 5.1 outputs `false`, go to the next triple (i_1, i_2, i_3). Otherwise go to the next α.
   (d) Adjoin i_1, i_2, i_3 to the sets T_1, T_2, T_3, respectively (counting multiplicities).
(4) If the size of each set T_1, T_2, T_3 is not equal to D, go to Step 2.
(5) (Computing the Igusa class polynomials.) For $i \in \{1, 2, 3\}$, do the following:
   (a) Compute $F_{i,p}(x) = \lambda_i \prod_{j \in T_i}(x - j)$ in $\mathbb{F}_p[x]$.
   (b) Use the Chinese Remainder Theorem to compute $F_i'(x) \in \mathbb{Z}[x]$ such that $F_i'(x) \equiv F_i(x) \pmod{B}$, $F_i'(x) \equiv F_{i,p}(x) \pmod{p}$, and the coefficients of $F_i'(x)$ are in the interval $[-pB/2, pB/2]$.
   (c) If $F_i'(x) = F_i(x)$, output $H_i(x) = \lambda_i^{-1} F_i(x)$. If $H_i(x)$ has been output for all i, terminate the algorithm.
   (d) Set $F_i(x) \leftarrow F_i'(x)$.
(6) Set $B \leftarrow pB$, and return to Step 2.

Proof. In view of [EL, Theorem 1], it suffices to prove that Step 3c correctly determines the set of curves with $\mathrm{End}(\mathrm{Jac}(C)) = \mathcal{O}_K$. It follows from Section 3 that $\mathrm{End}(\mathrm{Jac}(C)) = \mathcal{O}_K$ if and only if each of the elements of the generating set listed in Proposition 3.8 is an endomorphism.

By Algorithm 2.1, the π computed in Step 3b is such that π^σ is the Frobenius element of $\mathrm{Jac}(C)$ for some $\sigma \in \mathrm{Aut}(K/\mathbb{Q})$. By Corollary 3.10, $\mathrm{End}(\mathrm{Jac}(C)) = \mathcal{O}_K$ if and only if β^σ is an endomorphism for each β in the generating set of Proposition 3.8. Since elements of $\mathrm{Aut}(K/\mathbb{Q})$ preserve $\mathcal{O}_K$ as a set, $[\mathcal{O}_K : \mathbb{Z}[\pi^\sigma, \overline{\pi}^\sigma]] = [\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$.

For each ℓ dividing $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$, Steps 3(c)i and 3(c)ii test probabilistically whether $\frac{(\pi^\sigma)^k - 1}{\ell}$ is an endomorphism for an appropriate k. By Corollary 3.5, for any such ℓ dividing $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$ exactly, this suffices to determine whether $\frac{n}{\ell}\alpha^\sigma$ is an endomorphism for each $\alpha \in \mathcal{B} \setminus \mathbb{Z}$.

By Corollary 5.2, if ℓ^2 divides $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$ then Step 3(c)iii tests probabilistically whether $\frac{n}{\ell^d}\alpha^\sigma$ is an endomorphism. The input uses the field $\mathbb{F}_{p^{k'}}$ because

Proposition 6.3 implies that if the ℓ-torsion points are defined over $\mathbb{F}_{p^k}$, then the ℓ^d-torsion points are defined over $\mathbb{F}_{p^{k'}}$. □

Remark 7.2. Note that Step 5 differs from the corresponding step in [EL, Theorem 1]. Our version of the algorithm minimizes the amount of computation by terminating the algorithm in Step 5c as soon as the polynomials agree modulo two consecutive primes. For each prime p_i in an increasing sequence of primes, we compute a polynomial $F_{i,p}(x)$ that is congruent to the Igusa class polynomial $H_i(x)$ modulo the prime p_i [EL, Theorem 2]. We then use the Chinese Remainder Theorem and the collection of polynomials $\{F_{i,p}(x)\}$ to compute a polynomial $F_i(x)$ modulo $b_i = \prod_{j=1}^{i} p_j$. If $F_i(x) = F_{i+1}(x)$, then with high probability the coefficients of $H_i(x)$ are less than b_{i+1}, and thus $F_i(x)$ is equal to $H_i(x)$ itself. This conclusion is justified by the fact that if an integer n has the property that it is the same modulo b_i and modulo b_{i+1}, then $n = a_i + r_i b_i = a_{i+1} + r_{i+1} b_{i+1}$, with $a_i < b_i$ and $a_i = a_{i+1}$. It follows that p_{i+1} divides r_i. Since the probability of this happening for a random number r_i is $1/p_{i+1}$, the probability that all coefficients would simultaneously satisfy this congruence is $(1/p_{i+1})^{D+1}$, so most likely we have that actually $r_{i+1} = 0$ for each coefficient.

Remark 7.3. The λ_i input into the algorithm can be taken to be products of primes bounded in [GL], raised to a power that will be made explicit in forthcoming work. In practice, the power can be taken to be a small multiple of 6.

Since we check after every prime p_i whether the algorithm is finished, we do not need to know in advance the number of primes p_i that we will need to use. Thus the only bounds that need to be computed in advance are the bounds λ_i on the denominators of the coefficients of the Igusa class polynomials. In particular, we do not need to have a bound on either the numerators or the absolute values of the coefficients.

8. Implementation notes

Our most significant observation is that in practice, the running time of the probabilistic CRT algorithm is dominated by generating p^3 curves for each small p. Steps (3a) and (3b) of Algorithm 7.1 generate a list of curves C for which $\mathrm{End}(\mathrm{Jac}(C))$ is an order in $\mathcal{O}_K$. Algorithms 4.3 and 5.1 determine which endomorphism rings are equal to $\mathcal{O}_K$. Data comparing the relative speeds of these two parts of the algorithm appear in Section 9. This section describes a number of ways to speed up Algorithm 7.1, which are reflected in the running times that appear in Section 9.

(1) If p and k are large, then arithmetic on $J(\mathbb{F}_{p^k})$ is prohibitively slow, which slows down Algorithms 4.3 and 5.1. Since for various ℓ dividing the index $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$, the extension degrees k depend only on the prime p and the CM field K and not on the curve C, these extension degrees may be computed in advance (via Algorithm 4.1) before generating any curves. We set some bound N and tell the program that if the extension degree k for some ℓ is such that $p^k > N$, we should skip that p and go on to the next prime. For example, if $K = \mathbb{Q}(i\sqrt{13 + 2\sqrt{13}})$ and $p = 53$ (see Example 9.2), we have $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]] = 3^2 \cdot 43$, and the 43-torsion of a Jacobian J

with $\mathrm{End}(J) = \mathcal{O}_K$ will be defined over $\mathbb{F}_{p^{924}}$, a field of over 5000 bits that is far too large for our current implementation to handle efficiently.

(2) In a similar vein, since the speed of Algorithms 4.3 and 5.1 is determined by the size of the fields $\mathbb{F}_{p^k}$, for optimum performance one should perform these calculations in order of increasing k, so that as the fields get larger there are fewer curves to check.

(3) Algorithms 4.3 and 5.1 take a single curve as input. In Algorithm 7.1 those algorithms are executed with the same field K and many different curves, so any parameter that only depends on the field K and the prime p can be precomputed and stored for repeated reference. For example, the representation $\alpha = (a_0 + a_1\pi + a_2\pi^2 + a_3\pi^3)/n$ and the extension degrees k in Step 3(c)i can be computed only once. In addition, all of the curves that pass Step 3b have one of a small number of given zeta functions. Since $\#J(\mathbb{F}_{p^k})$ is determined by the zeta function, this number can also be computed in advance.

(4) If $\mathbb{F}_{p^k}$ is small enough, it may be faster to check fields of definition using the brute force method of Section 4.1, rather than Algorithm 4.3. If ℓ is small (as must be the case for k to be small), then we often find that $\#J(\mathbb{F}_{p^k}) = \ell^s m$ with $s \gg 4d$, and thus the number of random points needed in Algorithms 4.3 and 5.1 will be very large. While computing the group structure is an exponential-time computation, we find that if the group has size at most 2^{200}, MAGMA can compute the group structure fairly quickly.

(5) If Step 5c has already output $H_j(x)$ for some j, the roots of this polynomial mod p can be used as the possible values of i_j in Step 3. This will greatly speed up the calculation of the $F_{i,p}$ for the remaining primes: if one H_j has been output then only p^2D curves need to be computed (instead of p^3), and if two H_j have been output then only pD^2 curves need to be computed.

(6) In practice, for small primes p ($p < 800$ in our MAGMA implementation), computing $\#C(\mathbb{F}_p)$ (Step 5b of Algorithm 2.1) is more efficient than choosing a random point on $J(\mathbb{F}_p)$ and determining whether it is killed by one of the potential group orders (Step 5a of Algorithm 2.1), so these two steps should be switched for maximum speed. However, as p grows, the order of the steps as presented will be the fastest.

9. Examples

This section describes the performance of Algorithm 7.1 on three quartic CM fields: $\mathbb{Q}(i\sqrt{2+\sqrt{2}})$, $\mathbb{Q}(i\sqrt{13+2\sqrt{13}})$, and $\mathbb{Q}(i\sqrt{29+2\sqrt{29}})$. These fields are all Galois and have class number 1, so the density of primes with the desired splitting behavior is maximal. The Igusa polynomials are linear; they have integral coefficients for the first two fields, and have denominators dividing 5^{12} for the last. In all three examples, as p grows, the running time of the algorithm becomes dominated by the computation of p^3 curves for each p, whereas it was previously suspected that the the endomorphism ring computation would be the slow step in the CRT algorithm. A fast implementation in C to produce the curves from their Igusa invariants and to test the numbers of points would thus significantly improve the running time of the CRT algorithm.

Details of the algorithms' execution are given below. The algorithms were run on a 2.39 GHz AMD Opteron with 4 GB of RAM. The table headings have the following meaning:

- p: Size of prime field over which curves were generated.
- ℓ^d: Prime powers appearing in the denominators n of elements α input into Algorithms 4.3 and 5.1, when written in the form (3.2).
- k: Degrees of extension fields over which ℓ^d-torsion points are expected to be defined. These are listed in the same order as the corresponding ℓ^d.
- Curves: Time taken to generate p^3 curves and determine which have CM by K (cf. Algorithm 2.1).
- #Curves: Number of curves computed whose Jacobians have CM by K.
- 4.3 & 5.1: Time taken to run Algorithms 4.3 and 5.1 to find the single curve whose Jacobian has endomorphism ring equal to $\mathcal{O}_K$.

Example 9.1. We ran Algorithm 7.1 with $K = \mathbb{Q}(i\sqrt{2+\sqrt{2}})$ and $\lambda_1, \lambda_2, \lambda_3 = 1$. The results appear in Table 1. The last column of the table shows the intermediate polynomials $F_i(x)$ computed via the Chinese Remainder Theorem in Step 5b. The algorithm output the $F_i(x)$ listed for $p = 151$ as the Igusa class polynomials of K.

TABLE 1. Results for Algorithm 7.1 run with $K = \mathbb{Q}(i\sqrt{2+\sqrt{2}})$ and $\lambda_1, \lambda_2, \lambda_3 = 1$.

p	ℓ^d	k	Curves	#Curves	4.3 & 5.1	$F_i(x)$
7	2,4	2,4	0.5 sec	7	0.3 sec	$x+2$ $x+5$ $x+6$ (mod 7)
17	4,8	2,4	4 sec	39	0.2 sec	$x-54$ $x+19$ $x-8$ (mod 119)
23	2,4,7	2,4,3	9 sec	49	2.3 sec	$x+1017$ $x+852$ $x+111$ (mod 2737)
71	2,4	2,4	255 sec	7	0.7 sec	$x-75619$ $x+28222$ $x-46418$ (mod 194327)
97	4,8	2,4	680 sec	39	0.3 sec	$x-8237353$ $x+9355918$ $x+9086951$ (mod 18849719)
103	2,4,17	2,4,16	829 sec	119	17.6 sec	$x+104860961$ $x-28343520$ $x-9762768$ (mod 1941521057)
113	7,8,32	6,4,16	1334 sec	1281	28.8 sec	$x-1836660096$ $x-28343520$ $x-9762768$ (mod 219391879441)
151	2,4,7,17	2,4,6,16	0.2 sec	1	–	$x-1836660096$ $x-28343520$ $x-9762768$ (mod 33128173795591)

The total time of this run was 3162 seconds, or about 53 minutes. We observe that the polynomials F_2 and F_3 agree for $p = 103$ and $p = 113$. We deduce

that these polynomials are the correct Igusa polynomials, and following note (5) of Section 8, we use their roots for the values of i_2 and i_3 for $p = 151$. Thus instead of computing $151^3 \approx 2^{22}$ curves, we need to compute only 151 curves, out of which we can easily choose the right one. As a result, the computation for $p = 151$ takes practically no time at all. The same phenomenon also appears for the last prime in Examples 9.2 and 9.3.

Example 9.2. We ran Algorithm 7.1 with $K = \mathbb{Q}(i\sqrt{13 + 2\sqrt{13}})$ and $\lambda_1, \lambda_2, \lambda_3 = 1$. The results appear in Table 2. The algorithm output the following Igusa class polynomials:

$$x - 1836660096, \quad x - 28343520 \quad x - 9762768.$$

The total time of this run was 6969 seconds, or about 116 minutes. In this example we skip some primes because Algorithms 4.3 and 5.1 would need to compute in fields which are too large to be practical. In particular, for $p = 29, 53, 107, 139$, the algorithms would run over extension fields of degree $264, 924, 308, 162$, all of which have well over 1000 bits. Skipping these primes has no effect on the ultimate outcome of the algorithm.

TABLE 2. Results for Algorithm 7.1 with $K = \mathbb{Q}(i\sqrt{13 + 2\sqrt{13}})$ and $\lambda_1, \lambda_2, \lambda_3 = 1$.

p	ℓ^d	k	Curves	#Curves	4.3 & 5.1
29	3,23	2,264	–	–	–
53	3,43	2,924	–	–	–
61	3	2	167 sec	9	0.2 sec
79	27	18	376 sec	81	8.1 sec
107	9,43	6,308	–	–	–
113	3,53	1,52	1118 sec	159	137.2 sec
131	9,53	6,52	1872 sec	477	127.4 sec
139	9,243	6,162	–	–	–
157	9,81	6,54	3147 sec	243	16.5 sec
191	3,4,8	2,2,4	0.2 sec	1	–

Example 9.3. We ran Algorithm 7.1 with $K = \mathbb{Q}(i\sqrt{29 + 2\sqrt{29}})$ and $\lambda_1, \lambda_2, \lambda_3 = 5^{12}$. The results appear in Table 3. The algorithm output the following Igusa class polynomials:

$$x - \frac{2614061544410821165056}{5^{12}}, \quad x + \frac{586040972673024}{5^6}, \quad x + \frac{203047103102976}{5^6}.$$

The total time of this run was 56585 seconds, or about 15 hours, 43 minutes. In this example we again skip some primes because the fields input to Algorithms 4.3 and 5.1 would be too large. We also note that for $p = 7$, $\mathcal{O}_K = \mathbb{Z}[\pi, \overline{\pi}]$, so any curve over $\mathbb{F}_7$ that has a correct zeta function already has CM by all of $\mathcal{O}_K$, and we do not need to run Algorithms 4.3 and 5.1.

TABLE 3. Results for Algorithm 7.1 with $K = \mathbb{Q}(i\sqrt{29+2\sqrt{29}})$ and $\lambda_1, \lambda_2, \lambda_3 = 5^{12}$.

p	ℓ^d	k	Curves	#Curves	4.3 & 5.1
7	–	–	0.3 sec	1	–
23	13	84	9 sec	15	70.7 sec
53	7	6	105 sec	7	0.5 sec
59	4,5,8	2,12,4	164 sec	322	6.4 sec
83	3,5	4,24	431 sec	77	9.8 sec
103	67	1122	–	–	–
107	7,13	6,42	963 sec	105	69.3 sec
139	7,25	2,60	2189 sec	259	62.1 sec
181	9,27	6,18	84 min	161	3.6 sec
197	5,109	24,5940	–	–	–
199	25	60	106 min	37	1355.3 sec
223	4,8,23	2,4,22	174 min	1058	35.1 sec
227	109	1485	–	–	–
233	5,7,13	8,3,28	193 min	735	141.6 sec
239	7,109	6,297	–	–	–
257	3,7,13	4,6,84	286 min	1155	382.8 sec
277	5,7,23	24,6,22	0.3 sec	1	–

Remark 9.4. The data in Examples 9.1, 9.2, and 9.3 suggest that odd primes dividing the index $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$ always split in $\mathcal{O}_{K_0}$, the ring of integers of K_0. In fact the factorization of the index $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$ was given in [EL, Proposition 5] for primitive quartic CM fields K when K_0 has class number 1. We write $\pi = c_1 + c_2\sqrt{d} + (c_3 + c_4\sqrt{d})\eta$, where the c_i are rational numbers with only powers of 2 in the denominators and $\eta = i\sqrt{a + b\sqrt{d}}$ with $a, b, d \in \mathbb{Z}$, $d > 0$ and square-free. Then the index is, up to powers of 2, the product of c_2 with $(c_3^2 - c_4^2 d)$, where c_2 is the index of $\mathbb{Z}[\pi + \overline{\pi}]$ in $\mathcal{O}_{K_0}$ up to a power of 2. If a prime divides $(c_3^2 - c_4^2 d)$ exactly, i.e. the square of the prime does not divide it, then the prime splits in K_0. Thus primes different from 2 dividing the index $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$ exactly either split in K_0 or divide the index $[\mathcal{O}_{K_0} : \mathbb{Z}[\pi + \overline{\pi}]$. So except possibly for primes dividing c_2, no odd primes dividing the index $[\mathcal{O}_K : \mathbb{Z}[\pi, \overline{\pi}]]$ exactly are inert or totally ramified in K. If K is Galois, then this is enough to ensure that the extension degree k determined by Proposition 6.2 is at most ℓ^2. This agrees with the data in our examples, all of which considered Galois fields.

In practice, if a prime ℓ is inert or totally ramified in K, it would almost certainly be skipped anyway, since Proposition 6.2 shows that the ℓ-torsion may be defined over an extension field of degree $k \sim \ell^3$, which is too large to be practical (cf. Note (1) of Section 8). However the theoretical running times of Algorithms 4.3 and 5.1, given by Propositions 4.6 and 5.5 respectively, improve if inert or ramified primes ℓ are not considered. The slow step of both algorithms is computing a random point on $J(\mathbb{F}_{p^k})$, which takes roughly $O(k^2 \log k (\log p)^2)$ $\mathbb{F}_p$ operations. Since the bound on ℓ is p^2, if k is bounded by ℓ^2 instead of ℓ^3, this step would run in $O(p^8 \log^3 p)$ instead of $O(p^{12} \log^3 p)$ time.

References

[ALV] A. Agashe, K. Lauter, R. Venkatesan, "Constructing elliptic curves with a known number of points over a prime field," In *High Primes and Misdemeanours: lectures in honour of the 60th birthday of Hugh Cowie Williams*, Fields Institute Communications Series **42**, 2004, 1–17.

[AM] A.O.L. Atkin, F. Morain, "Elliptic curves and primality proving," *Math. Comp.* **61** (1993), 29–68.

[Ber] D.J.Bernstein, "Multidigit modular multiplication with the Explicit Chinese Remainder Theorem," Chapter 4, Ph.D. thesis, University of California at Berkeley, 1995.

[Brö] R. Bröker, "A p-adic algorithm to compute the Hilbert class polynomial," Preprint, 2006. `http://www.math.ucalgary.ca/~reinier/pub/padicj.pdf`

[CNST] J. Chao, O. Nakamura, K. Sobataka, S. Tsujii, "Construction of secure elliptic cryptosystems using CM tests and liftings," in *ASIACRYPT '98* Springer LNCS **1514**, Beijing, 1998, 95–109.

[Coh] H. Cohen, *A course in computational algebraic number theory*, Springer GTM **138**, 1993.

[CL] H. Cohn, K. Lauter, "Generating genus 2 curves with complex multiplication," Microsoft Research Internal Technical Report, 2001.

[Cou] J-M. Couveignes, "Linearizing torsion classes in the Picard group of algebraic curves over finite fields," Preprint, 2006. `http://www.picard.ups-tlse.fr/~couveig/publi/jaco.pdf`.

[CH] J.-M. Couveignes, T. Henocq, "Action of modular correspondences around CM-points," in *ANTS-V*, Springer LNCS **2369**, 2002, 234–243.

[CQ] G. Cardona, J. Quer, "Field of moduli and field of definition for curves of genus 2," in *Computational aspects of algebraic curves*, *Lecture Notes Ser. Comput.* **13**, World Sci. Publ., Hackensack, NJ, 2005, 71–83.

[EL] K. Eisenträger, K. Lauter, "A CRT algorithm for constructing genus 2 curves over finite fields," to appear in *AGCT-11*, 2007, `http://arxiv.org/abs/math.NT/0405305`.

[Eng] A. Enge, "The complexity of class polynomial computation via floating point approximations," Preprint, 2006. `http://fr.arxiv.org/abs/cs.CC/0601104`.

[GG] J. von zur Gathen, J. Gerhard, *Modern Computer Algebra*, 2nd ed., Cambridge University Press, Cambridge, 2003.

[GH] P. Gaudry, R. Harley, "Counting Points on Hyperelliptic Curves over Finite Fields," in *ANTS-IV*, Springer LNCS **1838**, 2000, 297–312.

[GHKRW] P. Gaudry, T. Houtmann, D. Kohel, C. Ritzenthaler, A. Weng, "The 2-adic CM method for genus 2 curves with application to cryptography," in *ASIACRYPT '06*, Springer LNCS **4284**, 2006, 114–129.

[GS] P. Gaudry and É. Schost, "Construction of secure random curves of genus 2 over prime fields," in *EUROCRYPT '04*, Springer LNCS **3027**, 2004, 239–256.

[GL] E. Goren, K. Lauter, "Class invariants for quartic CM fields," *Annales de l'Institut Fourier*, **57** (2007), 457–480.

[How] E. Howe, "Principally polarized ordinary abelian varieties over finite fields", *Trans. Amer. Math. Soc.* **347** (1995) 2361–2401.

[LLL] A.K. Lenstra, H.W. Lenstra, L. Lovász, "Factoring polynomials with rational coefficients," *Math. Ann.* **261** (1982), 515–534.

[LPP] H. W. Lenstra, Jr., J. Pila, C. Pomerance, "A hyperelliptic smoothness test II," *Proc. London Math. Soc.*, (3) **84** (2002), 105–146.

[Mes] J.-F. Mestre, "Construction de courbes de genre 2 à partir de leurs modules," in *Effective methods in algebraic geometry*, Birkhäuser Progr. Math. **94**, 1991, 313–334.

[Neu] J. Neukirch, *Algebraic Number Theory*, trans. Norbert Schappacher, Springer-Verlag, Berlin, 1999.

[Ros] S. Ross, *A First Course in Probability*, 5th ed., Prentice-Hall, Upper Saddle River, NJ, 1998.

[Sch] E. Schaefer, "A new proof for the non-degeneracy of the Frey-Rück pairing and a connection to isogenies over the base field," in *Computational aspects of algebraic curves*, *Lecture Notes Ser. Comput.* **13**, World Sci. Publ., Hackensack, NJ, 2005, 1–12.

[Shi] Goro Shimura. *Abelian varieties with complex multiplication and modular functions*, *Princeton Mathematical Series* **46**, Princeton University Press, Princeton, NJ, 1998.

[Spa] A.-M. Spallek, "Kurven vom Geschlecht 2 und ihre Anwendung in Public-Key-Kryptosystemen," Ph.D. thesis, Institut für Experimentelle Mathematik, Universität GH Essen, 1994.

[SW] B. K. Spearman, K. S. Williams, "Relative integral bases for quartic fields over quadratic subfields," *Acta Math. Hungar.* **70** (1996), 185–192.

[vW] P. van Wamelen, "Examples of genus two CM curves defined over the rationals," *Math. Comp.* **68** (1999), 307–320.

[W] A. Weng, "Constructing hyperelliptic curves of genus 2 suitable for cryptography," *Math. Comp.* **72** (2003), 435–458.

University of California, Berkeley, dfreeman@math.berkeley.edu

Microsoft Research, klauter@microsoft.com