LMS J. Comput. Math. 19 (Special issue A) (2016) 371–390 © 2016 Authors
doi:10.1112/S1461157016000358

Fast heuristic algorithms for computing relations in the class group of a quadratic order, with applications to isogeny evaluation

Jean-François Biasse, Claus Fieker and Michael J. Jacobson Jr

Abstract

In this paper, we present novel algorithms for finding small relations and ideal factorizations in the ideal class group of an order in an imaginary quadratic field, where both the norms of the prime ideals and the size of the coefficients involved are bounded. We show how our methods can be used to improve the computation of large-degree isogenies and endomorphism rings of elliptic curves defined over finite fields. For these problems, we obtain improved heuristic complexity results in almost all cases and significantly improved performance in practice. The speed-up is especially high in situations where the ideal class group can be computed in advance.

1. *Introduction*

Given an ideal $\mathfrak{a}$ in an order $\mathcal{O}$ in a quadratic field K and a bound $B > 0$, this paper presents heuristic techniques to decompose $\mathfrak{a} = (\alpha)\prod_i \mathfrak{p}_i^{e_i}$, where $\alpha \in K$, $\mathcal{N}(\mathfrak{p}_i) \leqslant B$ and where $\log|e_i| \leqslant \log|D|^{1/3} \log\log^{2/3}|D|$ for $D = \operatorname{disc}(\mathcal{O})$. Such a decomposition corresponds to the relation $\mathfrak{a} \sim \prod_i \mathfrak{p}_i^{e_i}$ in $\mathrm{Cl}(\mathcal{O})$. The search for relations between prime ideals of small norm has a direct application to the computation of $\mathrm{Cl}(\mathcal{O})$ in subexponential time. Ideal decomposition techniques are also used to solve the discrete logarithm problem in $\mathrm{Cl}(\mathcal{O})$ in subexponential time and the principal ideal problem (in real quadratic fields).

The main motivation for our ideal decomposition method is its application to algorithms for elliptic curves. In particular, it allows us to evaluate the action of an ideal $\mathfrak{a} \subseteq \mathcal{O}$ on the isomorphism class of a curve E over $\mathbb{F}_q$ satisfying $\mathrm{End}(E) \simeq \mathcal{O}$. Following the approach of the Schoof–Elkies–Atkin algorithm, standard techniques enable the evaluation of the action of a prime ideal [16]. These methods are impractical for ideals of large norm. In [4] Bröker, Charles, and Lauter presented a method relying on the decomposition of the input ideal over a factor basis of prime ideals of short norm. This strategy was subsequently rigorously analyzed by Jao and Soukharev [11] who proved that the run-time was subexponential. Ideal decomposition also applies to the computation of the endomorphism ring of an ordinary elliptic curve over $\mathbb{F}_q$. Bisson and Sutherland [3] showed how to calculate $\mathrm{End}(E)$ in subexponential time using random relations in $\mathrm{Cl}(\mathcal{O})$ for candidate rings with $\mathbb{Z}[\pi] \subseteq \mathcal{O} \subseteq \mathcal{O}_K$, where π_q is the Frobenius endomorphism.

In the context of applications to algorithms for elliptic curves, being able to find short relations (that is, relations with small exponents that involve only prime ideals of small norm) has a direct impact on the running time. Isogenies of degree equal to the norm of each prime ideal in the relation are evaluated, at a cost on the order of $O(\ell^3)$ for norm ℓ. This forces rather severe restrictions on the norm bound for finding relations required to optimize the overall asymptotic running time. In practice, the small norm bound makes relations much more difficult to find than is the case for other applications such as computing the ideal class group.

Received 22 February 2016.

2010 Mathematics Subject Classification 11R29 (primary), 11Y16, 11R11, 14K02 (secondary).

Contributed to the Twelfth Algorithmic Number Theory Symposium (ANTS-XII), Kaiserslautern, Germany, 29 August–2 September 2016.

Our ideal decomposition method relies on the assumption that $\mathrm{Cl}(\mathcal{O})$ is generated by a small number of ideal classes. We start by computing a first decomposition of the class of the input ideal $\mathfrak{a}$ over a larger set of primes $\mathcal{B}$ (which is computationally easier). Then we compute the lattice $\mathcal{L}$ of all relations between ideals of $\mathcal{B}$. By using the Hermite normal form (HNF) of $\mathcal{L}$, we deduce a decomposition of the class of $\mathfrak{a}$ over a small set of generators of $\mathrm{Cl}(\mathcal{O})$ (with large exponents). Finally, we solve the closest vector problem (CVP) in a low dimensional lattice to deduce a decomposition of the class of $\mathfrak{a}$ involving small exponents.

Contribution. We present a new heuristic technique for deriving short relations in $\mathrm{Cl}(\mathcal{O})$, and we apply it to the evaluation of an isogeny of large degree on an ordinary elliptic curve and to the computation of $\mathrm{End}(E)$.

Theoretical contributions

- We achieve a better asymptotic complexity for the decomposition of an input ideal $\mathfrak{a}$ in $\mathrm{Cl}(\mathcal{O})$ over ideals of small norm under Heuristic (H) introduced in § 5. Our complexity is significantly better when the maximal order in which $\mathcal{O}$ is contained has small discriminant.
- Our methods impact the complexity of the computation of $\mathrm{End}(E)$ and of the evaluation of isogenies under Heuristic (H).

Practical contributions

- We present an implementation of our ideal decomposition technique finding short relations significantly faster than Sutherland's SmoothRelation C code used in [**3**, **11**].
- Our implementation of the evaluation of large-degree isogenies and the computation of $\mathrm{End}(E)$ performs significantly better than [**3**, **11**], and can handle inputs of cryptographic size. It is particularly fast when the ideal class group of $\mathrm{Cl}(\mathcal{O})$ is precomputed, in which case individual isogenies can be evaluated even faster.

2. *Background*

Let $\mathcal{O}$ be an order in a number field K. We denote by $\mathrm{Cl}(\mathcal{O})$ the ideal class group of $\mathcal{O}$. In $\mathrm{Cl}(\mathcal{O})$, two fractional ideals $\mathfrak{a}, \mathfrak{b}$ are equivalent if there is $\alpha \in K$ such that $\mathfrak{b} = (\alpha)\mathfrak{a}$. We denote this by $\mathfrak{a} \sim \mathfrak{b}$. The class number of $\mathcal{O}$ is denoted by $h_{\mathcal{O}}$ and it satisfies $h_{\mathcal{O}} \leqslant |D| \log|D|$, where D is the discriminant of $\mathcal{O}$ (see [**7**, § 5.10.1]). We denote the norm of a fractional ideal $\mathfrak{a}$ of K by $\mathcal{N}(\mathfrak{a})$.

Let E_1, E_2 be two elliptic curves defined over $\mathbb{F}_q$. An isogeny $\phi : E_1 \to E_2$ is a non-constant rational map defined over $\mathbb{F}_q$ which is also a group homomorphism from E_1 to E_2. Two curves over $\mathbb{F}_q$ are isomorphic over $\overline{\mathbb{F}}_q$ if and only if they have the same j-invariant given by $j := 1728(4a^3/(4a^3 + 27b^2))$. There exists an ℓ-isogeny between representatives of two isomorphism classes with j-invariant j_1, j_2 if and only if $\Phi_\ell(j_1, j_2) = 0$, where $\Phi_\ell(X, Y)$ is the ℓth modular polynomial.

Let E be an elliptic curve defined over $\mathbb{F}_q$. We denote the ring of endomorphisms of E by $\mathrm{End}(E)$. For ordinary elliptic curves over a finite field, $\mathrm{End}(E)$ is an order in an imaginary quadratic field.

3. *Ideal class group computation*

Let $(\mathfrak{p}_i)_{i\leqslant k}$ for some $k > 0$ be a set of prime ideals that generates $\mathrm{Cl}(\mathcal{O})$. Our results on the computation of short relations in $\mathrm{Cl}(\mathcal{O})$ for a quadratic order $\mathcal{O}$ rely on the knowledge of a basis of the lattice of exponent vectors $(e_i)_{i\leqslant k}$ such that $\prod_{i\leqslant k} \mathfrak{p}_i^{e_i} \sim (1)$. Getting this basis is the bottleneck of the computation of $\mathrm{Cl}(\mathcal{O})$. In this section, we present updated results on the

https://doi.org/10.1112/S1461157016000358 Published online by Cambridge University Press

asymptotic complexity of the usual subexponential methods for computing $\mathrm{Cl}(\mathcal{O})$. The main results are Conjecture 1 and Proposition 3.1.

- The computation of $\mathrm{Cl}(\mathcal{O}_K)$ has complexity in $L_\Delta(1/2, 3/\sqrt{8}+o(1))$, where $\Delta = \mathrm{disc}(\mathcal{O}_K)$.
- The computation of $\mathrm{Cl}(\mathcal{O})$ from $\mathrm{Cl}(\mathcal{O}_K)$ has complexity in

$$L_\Delta(1/2, \sqrt{2}/2+o(1)) + L_f(1/3, \sqrt[3]{64/9}+o(1)) \quad \text{with } [\mathcal{O}_K : \mathcal{O}] = f^2,$$

where $L_N(a,b) := e^{b\log(N)^a \log(N)^{1-a}}$. The algorithm for the computation of $\mathrm{Cl}(\mathcal{O})$ takes $\mathcal{O}$ as input and outputs generators $\mathfrak{g}_i$ with their order d_i. It also returns a factor basis $\mathcal{B} = \{\mathfrak{p} \mid \mathcal{N}(\mathfrak{p}) \leqslant L_\Delta(1/2, 1/\sqrt{8})\}$ and a matrix H in HNF whose rows generate the lattice of $\vec{v}$ such that $\mathcal{B}^{\vec{v}} \sim (1)$. The matrix H has shape

$$H = \begin{bmatrix} h_{1,1} & 0 & \dots & 0 \\ \vdots & h_{2,2} & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ * & * & \dots & h_{k,k} \\ \hdotsfor{4} \\ (0) & \dots & \dots & (0) \end{bmatrix},$$

and it plays a crucial role in our algorithms for deriving short relations in $\mathrm{Cl}(\mathcal{O})$.

3.1. *Updated complexity of the computation of* $\mathrm{Cl}(\mathcal{O}_K)$

Under the Generalised Riemann Hypothesis (GRH), the algorithm for computing $\mathrm{Cl}(\mathcal{O}_K)$ with the best asymptotic complexity is due to Hafner and McCurley [9]. Its complexity is in $L_\Delta(1/2, \sqrt{2}+o(1))$, where $\Delta = \mathrm{disc}(\mathcal{O}_K)$. This procedure is classic, and we only give its high-level description here. Let $\mathcal{B} = \{\mathfrak{p} \mid \mathcal{N}(\mathfrak{p}) \leqslant B\}$ for some $B > 0$ and $n = |\mathcal{B}|$.

ALGORITHM 1. Computing $\mathrm{Cl}(\mathcal{O}_K)$ (high-level description).

```
Require: Maximal order $\mathcal{O}_K$ of the imaginary quadratic field $K$.
Ensure: Generators and relations for $\mathrm{Cl}(\mathcal{O}_K)$.
1: Find $m = n^{1+o(1)}$ vectors $\vec{v}_i$ such that $\mathcal{B}^{\vec{v}_i} \sim (1)$.
2: Let $M = (\vec{v}_i)_{i \leqslant m}$. Compute the HNF $H = \binom{H_1}{(0)}$ of $M$.
3: Compute the Smith normal form $S = \mathrm{diag}(d_j) = UH_1V$ of $H_1$.
4: return $\bigoplus_i \mathbb{Z}/d_i\mathbb{Z}$, $(\mathfrak{g}_i = \mathcal{B}^{\vec{V}_i})_{i \leqslant n}$, $H_1$.
```

In light of the new results that have appeared since [9], we can prove, under the GRH, that Algorithm 1 has a better complexity than $L_\Delta(1/2, \sqrt{2}+o(1))$. The proof of this updated run-time is outside the scope of this paper and will be developed in a future work. In the rest of the paper, we rely on Conjecture 1 to evaluate the complexity of Algorithm 1.

CONJECTURE 1 (GRH). Let $\mathcal{O}_K$ be a quadratic maximal order and let $\Delta = \mathrm{disc}(\mathcal{O}_K)$. The complexity of computing $\mathrm{Cl}(\mathcal{O}_K)$ is in $L_\Delta(1/2, 3/\sqrt{8}+o(1))$.

3.2. *From* $\mathrm{Cl}(\mathcal{O}_K)$ *to* $\mathrm{Cl}(\mathcal{O})$

The complexity of our methods depend on the discriminant of the maximal order $\mathcal{O}_K$ rather than the discriminant of $\mathcal{O}$. An important part of algorithm of Bisson and Sutherland is, given an order $\mathcal{O} \subseteq \mathcal{O}_K$, to efficiently find a generating set of relations for $\mathrm{Cl}(\mathcal{O})$. In this section, we show how to use a precomputed generating set of relations for $\mathrm{Cl}(\mathcal{O}_K)$ to find the relations

https://doi.org/10.1112/S1461157016000358 Published online by Cambridge University Press

of $\mathrm{Cl}(\mathcal{O})$. Our method directly derives from a result of Klüners and Pauli [12], and it is very efficient when the discriminant of $\mathcal{O}_K$ is significantly less than that of $\mathcal{O}$. The algorithm of Klüners and Pauli gives us $\mathrm{Cl}(\mathcal{O})$ from $\mathrm{Cl}(\mathcal{O}_K)$ by using the exact sequence

$$1 \to \mathcal{O}^* \to {\mathcal{O}_K}^* \to \bigoplus_{\mathfrak{p}|f} {\mathcal{O}_K}_{\mathfrak{p}}^* / \mathcal{O}_{\mathfrak{p}}^* \to \mathrm{Cl}(\mathcal{O}) \to \mathrm{Cl}(\mathcal{O}_K) \to 1,$$

where f is the conductor of $\mathcal{O}$ and $\mathcal{O}_{\mathfrak{p}}$ denotes the localization of $\mathcal{O}$ at $\mathfrak{p}$. This means that the knowledge of ${\mathcal{O}_K}^*$ and $\mathrm{Cl}(\mathcal{O}_K)$ (and of the factorization of f) gives us $\mathcal{O}^*$ and $\mathrm{Cl}(\mathcal{O})$.

Algorithm 2. Computing $\mathrm{Cl}(\mathcal{O})$ from $\mathrm{Cl}(\mathcal{O}_K)$ (high-level description).

```
Require: Order $\mathcal{O} \subseteq \mathcal{O}_K$ of conductor $f$, generators and relations for $\mathrm{Cl}(\mathcal{O}_K)$.
Ensure: Generators and relations for $\mathrm{Cl}(\mathcal{O})$.
1: Compute generators $(g_i)_{i \leqslant k}$ and a relation matrix $M_1 \in \mathbb{Z}^{k \times k}$ for $\bigoplus_{\mathfrak{p}|f} {\mathcal{O}_K}_{\mathfrak{p}}^*/\mathcal{O}_{\mathfrak{p}}^*$.
2: Let $(\mathfrak{g}_i)_{i \leqslant l}$ be generators of $\mathrm{Cl}(\mathcal{O}_K)$ and $d_i, \alpha_i$ such that $\mathfrak{g}_i^{d_i} = (\alpha_i)\mathcal{O}_K$. $M_2 \leftarrow \mathrm{diag}(d_i)$.
3: For each $\alpha_i$, find $\vec{v}_i$ such that $\overline{\alpha_i} = (g_j)_{j \leqslant k}^{\vec{v}_i}$, where $\overline{\alpha_i}$ is the image of $\alpha_i$ in $\bigoplus_{\mathfrak{p}|f} {\mathcal{O}_K}_{\mathfrak{p}}^*/\mathcal{O}_{\mathfrak{p}}^*$.
4: $M_3 \leftarrow (-\vec{v}_i)_{i \leqslant l}$. $M \leftarrow \begin{pmatrix} M_1 & M_3 \\ M_2 & (0) \end{pmatrix}$.
5: Let $G_1, \ldots, G_k \leftarrow g_1, \ldots, g_k$. $G_{k+1}, \ldots, G_{k+l} \leftarrow \mathfrak{g}_1, \ldots, \mathfrak{g}_l$.
6: return $(G_i)_{i \leqslant k+l}$, $M$.
```

Algorithm 2 yields a representation of $\mathrm{Cl}(\mathcal{O})$ in terms of relations involving the generators of $\bigoplus_{\mathfrak{p}|f} {\mathcal{O}_K}_{\mathfrak{p}}^*/\mathcal{O}_{\mathfrak{p}}^*$ and $\mathrm{Cl}(\mathcal{O}_K)$. To compute endomorphism rings, we need to be able to represent an ideal class in $\mathrm{Cl}(\mathcal{O})$ as a vector of exponents in $\mathbb{Z}^{k+l}$ over this set of generators. Algorithm 3 describes a method to solve this problem.

Algorithm 3. Finding the class of $\mathfrak{a} \subseteq \mathcal{O}$ in $\mathrm{Cl}(\mathcal{O})$.

```
Require: Ideal $\mathfrak{a}$ in $\mathcal{O} \subseteq \mathcal{O}_K$ of conductor $f$, generators $(G_i)_{i \leqslant k+l}$ and relations $M$ for $\mathrm{Cl}(\mathcal{O})$.
    Generators $(\mathfrak{g}_j)_{j \leqslant l}$ for $\mathrm{Cl}(\mathcal{O}_K)$.
Ensure: Vector $\vec{v} \in \mathbb{Z}^{l+k}$ such that $(G_i)_{i \leqslant k+l}^{\vec{v}}$ is the class of $\mathfrak{a}$.
1: Decompose $\mathfrak{a}$ in $\mathrm{Cl}(\mathcal{O}_K)$. Find $\alpha, (x_i)_{i \leqslant k}$ such that $\mathfrak{a} = (\alpha) \prod_i \mathfrak{g}_i^{x_i}$.
2: Decompose $\alpha$ in $\bigoplus_{\mathfrak{p}|f} {\mathcal{O}_K}_{\mathfrak{p}}^*/\mathcal{O}_{\mathfrak{p}}^*$. Find $(y_j)_{j \leqslant l}$ such that $\overline{\alpha} = \prod_j g_j^{y_j}$.
3: Let $\vec{v} = (x_1, \ldots, x_k, y_1, \ldots, y_l)$.
4: return $\vec{v}$.
```

The run-time of Algorithm 3 depends on the complexity of factoring the conductor f. We use the number field sieve [14] for this task. To date, its asymptotic complexity is conjectural.

Heuristic 1 (NFS). The complexity of factoring $N > 0$ with the number field sieve [14] is in $L_N(1/3, \sqrt[3]{64/9} + o(1))$.

Proposition 3.1 (GRH) + (NFS). *Given quadratic orders $\mathcal{O} \subseteq \mathcal{O}_K$, $\mathrm{Cl}(\mathcal{O}_K)$, together with the HNF of the matrix corresponding to the relations between elements in $\mathcal{B} = \{\mathfrak{p} \mid \mathcal{N}(\mathfrak{p}) \leqslant L_\Delta(1/2, 1/\sqrt{8})\}$, the run-time of Algorithm 3 is in*

$$\log(\mathcal{N}(\mathfrak{a}))^{1+o(1)} + L_\Delta(1/2, \sqrt{2}/2 + o(1)) + L_f(1/3, \sqrt[3]{64/9} + o(1)) \quad \textit{with } [\mathcal{O}_K : \mathcal{O}] = f^2,$$

while the complexity of Algorithm 2 is in $L_\Delta(1/2, \sqrt{2}/2 + o(1)) + L_f(1/3, \sqrt[3]{64/9} + o(1))$.

Proof. The two bottlenecks of this computation are the factorization of f and the original decomposition of $\mathfrak{a}$ with respect to $\mathcal{B}$ in $\mathrm{Cl}(\mathcal{O}_K)$. Step 1 of Algorithm 2 is achieved using

[12, Algorithm 8.1], which is dominated by the factorization of f. Step 3 of Algorithm 2 consists of solving the discrete logarithm problem in $(\mathcal{O}/\mathfrak{p})^*$ for $\mathfrak{p} \mid f$. Decomposing $\mathfrak{a}$ over $\mathcal{B}$ is achieved by multiplying $\mathfrak{a}$ by random products of ideals in $\mathcal{B}$, reducing the result and testing if it is $\mathcal{B}$-smooth. Reducing $\mathfrak{a}$ induces the term $\log(\mathcal{N}(\mathfrak{a}))^{1+o(1)}$, while finding a smooth value takes time $L_\Delta(1/2, \sqrt{2}/2 + o(1))$. □

4. *Applications of the computation of short relations in* $\mathrm{Cl}(\mathcal{O})$

Action of ideals in $\mathrm{End}(E)$. Let $\mathcal{O}$ be a quadratic order and E be an ordinary elliptic curve over $\mathbb{F}_q$ with $\mathcal{O} \simeq \mathrm{End}(E)$. The class group of $\mathrm{End}(E)$ acts transitively on isomorphism classes of ordinary elliptic curves (that is, on j-invariants of curves) having the same endomorphism ring as E. More precisely, the class of an ideal $\mathfrak{a} \subseteq \mathcal{O}$ acts on the isomorphism class of E via an isogeny of degree $\mathcal{N}(\mathfrak{a})$. Let $\ell > 0$, and assume that we want to evaluate an isogeny of degree ℓ at $P \in E$. We choose an ideal $\mathfrak{a} \subseteq \mathcal{O}$ of norm ℓ and use the methods described in [4, § 3.1] to compute the isogeny φ of degree ℓ corresponding to the action of $\mathfrak{a}$. This method is attributed to Atkin and Elkies and was first published by Schoof [16, § 7, 8] in the context of point counting.

Bröker, Charles and Lauter [4] proposed an algorithm to evaluate a horizontal isogeny (that is, between curves of same endomorphism ring) of large degree ℓ at a point P on an ordinary curve E of trace t defined over $\mathbb{F}_p$. It produces a curve E' that is ℓ-isogenous to E and $\varphi(P)$, where $\varphi : E \to E'$ is the degree-ℓ isogeny between E and E'. It was subsequently rigorously analyzed by Jao and Soukharev [11]. Let $\mathfrak{a} \subseteq \mathcal{O}$ be an ideal satisfying $\mathcal{N}(\mathfrak{a}) = \ell$. The strategy of [4] consists in finding prime ideals $\mathfrak{p}_1, \ldots, \mathfrak{p}_k$ of small norm such that $\prod_i \mathfrak{p}_i$ is in the same ideal class as $\mathfrak{a}$. The Atkin–Elkies method would be too expensive if applied directly to $\mathfrak{a}$ and P. Instead, we apply it to the $(\mathfrak{p}_i)_{i \leqslant k}$ (which have small norm). This leads to the calculation of E_c in the same isomorphism class as E' and $\varphi_c(P)$, where $\varphi : E \to E_c$ is an isogeny. Then we apply the right isomorphism ι between E_c and E' to compute $\varphi(P)$.

Computing $\mathrm{End}(E)$. Let E be an ordinary elliptic curve of trace t over $\mathbb{F}_p$ and let π be its Frobenius endomorphism. The order $\mathbb{Z}[\pi]$ lies inside the quadratic field $K = \mathbb{Q}(\sqrt{t^2 - 4p})$. Let $\mathcal{O}_K$ be the maximal order of K and let $\mathcal{O}$ be the order isomorphic to $\mathrm{End}(E)$. We know that $\mathbb{Z}[\pi] \subseteq \mathcal{O} \subseteq \mathcal{O}_K$ and computing $\mathrm{End}(E)$ can be done by identifying the conductor u of $\mathcal{O}$. We follow the approach of Bisson and Sutherland [3] that relies on the following lemma.

LEMMA 4.1. *Let f be the conductor of $\mathbb{Z}[\pi]$, let u be the conductor of $\mathcal{O} \simeq \mathrm{End}(E)$, let Δ be the discriminant of $\mathcal{O}_K$ and let p^k be a prime power dividing f. Let $\mathcal{O}_1$ be the quadratic order of discriminant $D_1 = (f/p^j)^2\Delta$ and $\mathcal{O}_2$ be the quadratic order of discriminant $D_2 = p^{2k}\Delta$ for $j = v_p(f) - k + 1$. If a relation $\mathcal{R}$ of the form $\prod_i \mathfrak{p}_i \sim (1)$ holds in $\mathrm{Cl}(\mathcal{O}_1)$ but not in $\mathrm{Cl}(\mathcal{O}_2)$, then $p^k \mid u$ if and only if $\mathcal{R}$ does not hold in $\mathrm{Cl}(\mathcal{O})$.*

The strategy from [3] consists of factoring the conductor v and testing whether $p^k \mid u$ for $p \mid f$ and $k \leqslant v_p(f)$. To perform this test, we first find a relation of the form $\prod_i \mathfrak{p}_i \sim (1)$ holding $\mathrm{Cl}(\mathcal{O}_1)$ but not in $\mathrm{Cl}(\mathcal{O}_2)$. Then we test whether this relation holds in $\mathrm{Cl}(\mathcal{O})$, using the methods of [4, § 3.1].

5. *Finding short relations in* $\mathrm{Cl}(\mathcal{O})$

Given generators and relations for $\mathrm{Cl}(\mathcal{O})$, the question of finding short relations between classes of ideals of small norm is central to endomorphism ring computation and isogeny evaluation algorithms. In this section, we show that there is a basis of short relations for $\mathrm{Cl}(\mathcal{O})$ and

https://doi.org/10.1112/S1461157016000358 Published online by Cambridge University Press

we show how to compute it. To achieve this goal, we restrict ourselves to instances of our problem such that only $C\log^{2/3}(D)$ ideals generate $\mathrm{Cl}(\mathcal{O})$, where D is the discriminant of $\mathcal{O}$ and $C>0$ is a constant. The overwhelming majority of instances of our problems satisfy this condition. However, the best proven (under the GRH) bound, due to Bach, stipulates that $\mathrm{Cl}(\mathcal{O})$ is generated by the primes of norm less than $12\log^2(|D|)$. In the rest of the paper, we restrict our complexity analysis to the case where few ideals are required to generate $\mathrm{Cl}(\mathcal{O})$. However, our algorithms are valid for all $\mathcal{O}$ (with no guarantee on the run-time).

Definition 1 (Class $\mathcal{Q}_C$). Let $C>0$. We denote by $\mathcal{Q}_C$ the class of quadratic orders $\mathcal{O}$ such that $\mathrm{Cl}(\mathcal{O})$ is generated by at most $C\log^{2/3}(|D|)$ split primes, where $D=\mathrm{disc}(\mathcal{O})$.

A given quadratic order is in $\mathcal{Q}_C$ with overwhelming probability for a reasonable value of C (say, $C\geqslant 1$). The proper evaluation of the proportion of quadratic ideals belonging to $\mathcal{Q}_C$ for a given C is outside the scope of this paper. To illustrate numerically that few ideals are required to generate $\mathrm{Cl}(\mathcal{O}_K)$, we drew 100 random fundamental discriminants Δ of bit size $30, 50, \ldots, 150$. For each size, we reported in Table 1 the maximum number of split primes required to generate $\mathrm{Cl}(\mathcal{O}_K)$ over the 100 fundamental discriminants.

5.1. *Random walks in the Cayley graph of* $\mathrm{Cl}(\mathcal{O})$

Let $C>0$ be a constant. Suppose we are given a family of orders $\mathcal{O}$ in $\mathcal{Q}_C$. We need to argue that each ideal class in $\mathrm{Cl}(\mathcal{O})$ can be represented by an ideal with a short decomposition. We assume Heuristic 2.

Heuristic 2 (H). Let $C>0$, $\mathcal{O}\in\mathcal{Q}_C$ and $(\mathfrak{p}_i)_{i\leqslant k}$ be $C\log^{2/3}(|D|)$ split prime ideals that generate $\mathrm{Cl}(\mathcal{O})$. Then each class of $\mathrm{Cl}(\mathcal{O})$ has a representative of the form $\prod_i \mathfrak{p}_i^{n_i}$ for $|n_i|\leqslant e^{\log^{1/3}|D|}$.

The main assumption behind Heuristic 2 is that the total number $|D|\gg h_{\mathcal{O}}$ of different possible products provides generators for all classes in $\mathrm{Cl}(\mathcal{O})$. To illustrate this, we drew fundamental discriminants having between 20 and 50 decimal discriminants. For each of them, we computed the ideal class group and drew 1000 classes at random, then decomposed them over the set of the first $\log^{2/3}|D|$ invertible primes (using the short decomposition technique described in § 6). We reported the maximal absolute value of a coefficient occurring in a decomposition. The results presented in Table 2 show that this maximal value is systematically significantly less than $e^{\log^{1/3}|D|}$.

To prove Heuristic 1, the natural direction would be to follow the approach of Jao *et al.* [10]. They were able to give bounds on the eigenvalue of the Cayley graph $(S_x, \mathrm{Cl}(\mathcal{O}))$, where S_x is the set of primes $\mathfrak{p}$ with $\mathcal{N}(\mathfrak{p})\leqslant x$ together with their inverses. When $x>\log^2|D|$,

Table 1. *Maximal number of primes required to generate* $\mathrm{Cl}(\mathcal{O}_K)$.

$\log_2\lvert\Delta\rvert$	$\lfloor\log^{2/3}\lvert\Delta\rvert\rfloor$	Maximal number of primes
30	7	4
50	10	5
70	13	3
90	15	4
110	17	4
130	20	4
150	22	3

https://doi.org/10.1112/S1461157016000358 Published online by Cambridge University Press

a non-trivial inequality between the non-trivial eigenvalues λ and the trivial one λ_{triv} derives from [10, Theorem 1.1]. Then [10, Lemma 2.1] shows that random walks in $(S_x, \mathrm{Cl}(\mathcal{O}))$ of length $(\log(2|\mathrm{Cl}(\mathcal{O})|)/|S_x|^{1/2})/\log(\lambda_{\mathrm{triv}}/\lambda)$ have the same probability of ending in any subset of $\mathrm{Cl}(\mathcal{O})$. This proves that short decomposition of ideal classes exist when we allow all the primes of norm less than $\log^{2+\varepsilon}|D|$ to occur in the relations. With our restriction on the factor basis, the statement remains conjectural.

5.2. *Existence of short relations*

Let $C > 0$ and let $\mathcal{O}$ be a quadratic order in $\mathcal{Q}_C$ of discriminant D. Let $\mathcal{B} = \{\mathfrak{p}_i\}_{i \leqslant k}$ with $k \leqslant C \log^{2/3}(|D|)$ be a generating set for $\mathrm{Cl}(\mathcal{O})$. We show that there is a basis $b_1, \ldots, b_k$ of the lattice of relations between elements of $\mathcal{B}$ such that each $b_i \in \mathbb{Z}^k$ has entries with absolute value bounded by $e^{(1+o(1))\log^{1/3}|D|}$. We follow the approach of Bisson [2], which consists of rewriting relations between ideals of $\mathcal{B}$ and generators of $\mathrm{Cl}(\mathcal{O})$ with respect to short decompositions obtained from Corollary 2. For each invertible fractional ideal $\mathfrak{a}$, we define $\sigma(\mathfrak{a}) \in \mathbb{Z}^k$ to be one of the vectors $\vec{v}$ with entries bounded by $e^{\log^{1/3}|D|}$ such that $\mathfrak{a} \sim \mathcal{B}^{\vec{v}}$ (such a vector exists under Heuristic 2.

PROPOSITION 5.1 (H). *There is a basis of the lattice $\mathcal{L}$ of vectors $\vec{v}$ such that $\mathcal{B}^{\vec{v}} \sim (1)$ consisting of vectors with entries bounded by $e^{(1+o(1))\log^{1/3}|D|}$.*

Proof. Let $(\mathfrak{g}_i)_{i \leqslant l}$ be the generators of $\mathrm{Cl}(\mathcal{O})$ and d_i be the exponents such that $\mathfrak{g}_i^{d_i} \sim (1)$. The following vectors are in $\mathcal{L}$.

- $\sigma(\mathfrak{g}_i^{2^j}) - 2\sigma(\mathfrak{g}_i^{2^{j-1}})$ for $j \leqslant \log_2(d_i)$.
- $\sum_j b_j \sigma(\mathfrak{g}_i^{2^j})$, where b_j is the jth bit of d_i.

This is just rewriting that $\mathfrak{g}_i^{d_i} \sim (1)$. The class of each $\mathfrak{p} \in \mathcal{B}$ can be re-written with respect to the $\mathfrak{g}_i$ as $\mathfrak{p} \sim \prod \mathfrak{g}_i^{n_i}$. Then we have two short vectors representing $\mathfrak{p}$: $\sigma(\mathfrak{p})$ and $\sum_i \sum_j c_{i,j}\sigma(\mathfrak{g}_i^{2^j})$, where $c_{i,j}$ is the jth bit of d_i. The vectors $\sigma(\mathfrak{p}) - \sum_i \sum_j c_{i,j}\sigma(\mathfrak{g}_i^{2^j})$ are a basis for $\mathcal{L}$ and their coefficients are in $e^{(1+o(1))\log^{1/3}|D|}$. □

5.3. *Finding a short relation*

In this section, we show how to find short relations. Let $C > 0$ and let $\mathcal{O}$ be a quadratic order in $\mathcal{Q}_C$ of discriminant D. Let $\mathcal{B} = \{\mathfrak{p}_i\}_{i \leqslant k}$ with $k \leqslant C \log^{2/3}(|D|)$ be a generating set for $\mathrm{Cl}(\mathcal{O})$. We start by finding an arbitrary basis for the lattice $\mathcal{L}$ of relations between elements in $\mathcal{B} = \{\mathfrak{p}_i\}_{i \leqslant k}$ with $k \leqslant C \log^{2/3}(|D|)$. We assume that $\mathrm{Cl}(\mathcal{O}_K)$ was precomputed. Our method simply consists of repeated calls to Algorithm 3.

TABLE 2. *Maximal exponent occurring in short decompositions* (*over* 1000 *random instances*).

$\log_{10}\|D\|$	$\log^{2/3}\|D\|$	Maximal coefficient	$e^{\log^{1/3}\|D\|}$
20	13	6	36
25	15	8	48
30	17	7	61
35	19	9	75
40	20	10	91
45	22	14	110
50	24	13	130

https://doi.org/10.1112/S1461157016000358 Published online by Cambridge University Press

Algorithm 4. Relation matrix for $\mathcal{B}$.

```
Require: Order $\mathcal{O}$, generators and relations for $\mathrm{Cl}(\mathcal{O}_K)$ and $\mathcal{B} = \{\mathfrak{p}_i\}_{i \leqslant k}$.
Ensure: A basis for the lattice of relations between elements of $\mathcal{B}$ in $\mathrm{Cl}(\mathcal{O})$.
1: For each $\mathfrak{p}$ in $\mathcal{B}$, compute the class of $\mathfrak{p}$ with Algorithm 3.
2: Build the matrix of the morphism $\phi : \mathbb{Z}^k \to \mathrm{Cl}(\mathcal{O})$, defined by $\phi(\vec{v}) = [\mathcal{B}^{\vec{v}}]$.
3: Find a basis $b_1, \ldots, b_k$ of the kernel of $\phi$.
4: return $b_1, \ldots, b_k$.
```

Once a basis $b_1, \ldots, b_k$ of $\mathcal{L}$ is found, we use standard methods to compute $U \in \mathrm{GL}_n(\mathbb{Z})$ such that $U \cdot M = H$, where H is the HNF of M. The upper $k \times k$ block of the HNF of M has the shape $\left(\begin{smallmatrix} H_1 & (0) \\ H_2 & I \end{smallmatrix}\right)$, where I is an identity block. The block H_1 is called the essential part of the HNF. Under GRH, the dimensions of H_1 are in $O(\log^2|D|)$ while, under Hypothesis 2, they are in $O(\log^{2/3}|D|)$. This means that the rows of H_1 are relations with very short support. Unfortunately the coefficients of H_1 are of the order of magnitude of $\sqrt{|D|}$. However, we showed heuristically, in § 5.2, that the lattice $\mathcal{L}$ generated by the rows of H_1 contained short relations. To find them, we use the BKZ lattice reduction method. With the appropriate block size, its run-time is in $L_D(1/3, O(1))$.

Algorithm 5. Finding small relations in $\mathrm{Cl}(\mathcal{O})$.

```
Require: $C > 0$, $r > 0$, $\mathcal{O} \in \mathcal{Q}_D$ of discriminant $D$, $B > 0$, generators and relations for $\mathrm{Cl}(\mathcal{O}_K)$.
Ensure: A matrix $M'$ whose rows $(e_1, \ldots, e_k)$ are such that $\prod_{i \leqslant k} \mathfrak{p}_i^{e_i} \sim (1)$ for split primes $\mathfrak{p}_i$ generating $\mathrm{Cl}(\mathcal{O})$ and $k = C \log^{2/3}(|D|)$.
1: Let $\mathfrak{p}_1, \ldots, \mathfrak{p}_k$ be a generating set for $\mathrm{Cl}(\mathcal{O})$ for $k = C \log^{2/3}(|D|)$.
2: Let $\mathcal{B} = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_m\}$, where $\{\mathfrak{p}_{k+1}, \ldots, \mathfrak{p}_m\}$ are the split primes of norm less than $B$.
3: Find the matrix $M$ whose rows generate the lattice of relations between prime of $\mathcal{B}$ using Algorithm 4.
4: Find the HNF $\begin{pmatrix} H_1 & (0) \\ H_2 & I \end{pmatrix}$ of $M$, where $H_1 \in \mathbb{Z}^{k \times k}$.
5: Let $M'$ be the output of the BKZ reduction of the rows of $H_1$.
6: return $M'$.
```

Proposition 5.2 (GRH) + (NFS) + (H). *Given $\mathrm{Cl}(\mathcal{O}_K)$ and relations between primes in $\{\mathfrak{p} \mid \mathcal{N}(\mathfrak{p}) \leqslant L_\Delta(1/2, 1/\sqrt{8})\}$, Algorithm 5 used with block size $r = \log|D|^{1/3} \log\log^{1/2}|D|$ returns a basis of vectors of $\mathcal{L}$ with entries in $L_D(1/3, 1 + o(1))$ in time*

$$L_\Delta(1/2, \sqrt{2}/2 + o(1)) + L_D(1/3, 1 + o(1)) + L_f(1/3, \sqrt[3]{64/9} + o(1)) \quad \textit{with } [\mathcal{O}_K : \mathcal{O}] = f^2.$$

Proof. Ajtai, Kumar and Sivakumar's updated BKZ reduction [1] stipulates that we can obtain an approximation factor of $r^{k/r}$ in time $2^{O(r)}$. With $r = \log|D|^{1/3} \log\log^{1/2}|D|$, the run-time satisfies $2^{O(r)} \ll L_D(1/3, 1 + o(1))$. The rest of the run-time is bounded by Step 2 which is done using Algorithm 3 for all ideals in $\mathcal{B}$. The approximation factor $r^{k/r} = e^{\log(r)k/r}$ satisfies the inequality

$$\frac{k}{r}\log(r) \leqslant \frac{C}{3}\log^{1/3}(|D|)\log^{1/2}(|D|)(1 + o(1)) \ll \log^{1/3}(|D|)\log^{2/3}(|D|)(1 + o(1)).$$

Under Heuristic 2, the size of the smallest vector of $\mathcal{L}$ is no more than $e^{(1+o(1))\log^{1/3}|D|}$. Therefore, the entries of the output of Algorithm 5 are in $L_D(1/3, 1 + o(1))$. □

https://doi.org/10.1112/S1461157016000358 Published online by Cambridge University Press

6. *Computation of a short decomposition of an ideal*

In this section, we show how to compute a short decomposition of an input ideal $\mathfrak{a} \subseteq \mathcal{O}$. More specifically, we show that given $\mathrm{Cl}(\mathcal{O}_K)$ (together with a relation matrix in HNF form), an input ideal $\mathfrak{a}$ and a factor basis $\mathcal{B}$ such that $|\mathcal{B}| \leqslant C\log^{2/3}|D|$, where $D = \mathrm{disc}(\mathcal{O})$, our algorithm returns $(n_i)_{i\leqslant k}$ with $|n_i| \in L_D(1/3, 1+o(1))$ such that $\mathfrak{a} \sim \prod_i \mathfrak{p}_i^{n_i}$. This is used in § 7 to evaluate an isogeny of large degree $\ell = \mathcal{N}(\mathfrak{a})$. Our strategy has two main steps.

(1) Find a decomposition $\mathfrak{a} \sim \mathfrak{p}_1^{m_1} \ldots \mathfrak{p}_k^{m_k}$ in $\mathrm{Cl}(\mathcal{O})$, where $\mathcal{B} = \{\mathfrak{p}_i\}_{i\leqslant k}$.

(2) Refine the previous decomposition to $\mathfrak{a} \sim \mathfrak{p}_1^{n_1} \ldots \mathfrak{p}_k^{n_k}$, where $|n_i| \in L_D(1/3, 1+o(1))$.

6.1. *Description of the ideal decomposition procedure*

First decomposition. We decompose $\mathfrak{a}$ with respect to the factor basis of $\mathrm{Cl}(\mathcal{O})$ by calling Algorithm 3 and solving a linear system. The original decomposition has coefficients that can be as large as $|D|^{1/2}$, which prohibits efficient isogeny evaluation.

ALGORITHM 6. Decomposition of $\mathfrak{a}$ with respect to $\mathcal{B}$.

```
Require: $C > 0$, $\mathcal{O} \in \mathcal{Q}_C$, ideal $\mathfrak{a} \subseteq \mathcal{O}$, generators $(G_i)_{i\leqslant k+l}$ and relations for $\mathrm{Cl}(\mathcal{O})$,
    generating set $\mathcal{B} = \{\mathfrak{p}_i\}_{i\leqslant k}$ for $\mathrm{Cl}(\mathcal{O})$ with $k \leqslant C\log^{2/3}|D|$, $D = \mathrm{disc}(\mathcal{O})$.
Ensure: $(m_i)_{i\leqslant k}$ such that $\mathfrak{a} \sim \prod_i \mathfrak{p}_i^{m_i}$ in $\mathrm{Cl}(\mathcal{O})$.
Find the vector $\vec{a}$ of the decomposition of $\mathfrak{a}$ with respect to $(G_i)_{i\leqslant k+l}$ with Algorithm 3.
Find the vectors $\vec{v}_j$ of the decomposition the $\mathfrak{p}_j$ with respect to $(G_i)_{i\leqslant k+l}$ with Algorithm 3.
Let $M = (\vec{v}_j)_{j\leqslant k}$. Solve the linear system $\vec{m}M = \vec{a}$.
return $\vec{m}$.
```

PROPOSITION 6.1 (GRH) + (NFS) + (H). *Algorithm 6 returns a valid decomposition of $\mathfrak{a}$ with respect to $\mathcal{B} = \{\mathfrak{p}_i\}_{i\leqslant k}$ with $k \leqslant C\log^{2/3}|D|$, where $D = \mathrm{disc}(\mathcal{O})$ in time*

$$\log(\mathcal{N}(\mathfrak{a}))^{1+o(1)} + L_\Delta(1/2, \sqrt{2}/2 + o(1)) + L_f(1/3, \sqrt[3]{64/9} + o(1)) \quad \textit{with } [\mathcal{O}_K : \mathcal{O}] = f^2.$$

Refinement of the decomposition. The next step consists of finding a close vector $(e_i)_{i\leqslant k}$ to $(m_i)_{i\leqslant k}$ that belongs to the lattice of relations between elements of $\mathcal{B}$ and which satisfies $|\mathcal{B}| \leqslant C\log^{2/3}|D|$. Exactly solving the CVP is a hard problem. Even with a subexponential approximation factor, the dimension of the relation lattice (under GRH) is too large to allow for a subexponential run- time. In Proposition 6.2, we show that Algorithm 6 has a subexponential run-time if $\mathcal{O} \in \mathcal{Q}_C$ for a constant $C > 0$ and if we assume Heuristic 2.

ALGORITHM 7. Short decomposition of $\mathfrak{a}$ with respect to $\mathcal{B}$.

```
Require: $C > 0$, $\mathcal{O} \in \mathcal{Q}_C$ with $D = \mathrm{disc}(\mathcal{O})$, prime ideals $(\mathfrak{p}_i)_{i\leqslant k}$ with $k \leqslant C\log^{2/3}(|D|)$ that
    generate $\mathrm{Cl}(\mathcal{O})$ and a basis for the lattice $\mathcal{L}$ of $(v_i)_{i\leqslant k} \in \mathbb{Z}^k$ such that $\prod_{i\leqslant k} \mathfrak{p}_i^{v_i} \sim (1)$.
Ensure: Decomposition $\mathfrak{a} \sim \prod_i \mathfrak{p}_i^{n_i}$ in $\mathrm{Cl}(\mathcal{O})$ with $|n_i| \in L_D(1/3, 1+o(1))$.
Find a decomposition $\mathfrak{a} \sim \prod_i \mathfrak{p}_i^{m_i}$ in $\mathrm{Cl}(\mathcal{O})$ with Algorithm 6.
Use BKZ with block size $r = \log|D|^{1/3}\log\log^{2/3}|D|$ to get a reduced basis $(b_i)_{i\leqslant k}$ of $\mathcal{L}$.
Use Babai's nearest plane algorithm to find a close vector $(e_i)_{i\leqslant k}$ to $(m_i)_{i\leqslant k}$ in $\mathcal{L}$.
For $i \leqslant k$, $n_i \leftarrow m_i - e_i$.
return $(n_i)_{i\leqslant k}$.
```

https://doi.org/10.1112/S1461157016000358 Published online by Cambridge University Press

PROPOSITION 6.2 (GRH) + (NFS) + (H). *Algorithm 7 returns a valid decomposition of $\mathfrak{a}$ with respect to a factor basis $\mathcal{B}$ satisfying $|\mathcal{B}| \leqslant C \log^{2/3}|D|$ with coefficients in $L_D(1/3, 1+o(1))$ in time*

$$\log(\mathcal{N}(\mathfrak{a}))^{1+o(1)} + L_\Delta(1/2, \sqrt{2}/2 + o(1)) + L_D(1/3, 1 + o(1)) + L_f(1/3, \sqrt[3]{64/9} + o(1)),$$

with $[\mathcal{O}_K : \mathcal{O}] = f^2$.

Proof. The run-time of Step 1 is given by Proposition 6.1. The rest of the run-time of Algorithm 7 is dominated by Step 2. As we assume Heuristic 2, the dimension of the input lattice $\mathcal{L}$ is in $O(\log^{2/3}|D|)$, whereas it would be in $O(\log^2(|D|))$ if we were only assuming the GRH. This has a direct impact on the run-time of Step 2 of Algorithm 7. Apart from the dimension of $\mathcal{L}$, the run-time of Step 2 is ruled by the block size that is used to run the BKZ algorithm. Choosing $r = \log|D|^{1/3} \log\log^{1/2}|D|$ implies that the output of Step 2 is a reduced basis of $\mathcal{L}$ with vectors of size within a factor $r^{k/r}$ of the size of the shortest vectors of $\mathcal{L}$. With this block size, the run-time of Step 2 is in $2^{O(r)} \ll L_D(1/3, 1 + o(1))$.

Step 3 solves the approximate CVP problem, that is, it finds a vector $\vec{e} = (e_i)_{i\leqslant k} \in \mathcal{L}$ such that $\|\vec{e} - \vec{m}\| \leqslant \gamma \operatorname{dist}(\vec{m}, \mathcal{L})$ for some $\gamma \geqslant 1$, where $\vec{m} = (m_i)_{i\leqslant k}$. If the input of Step 3 is a basis for $\mathcal{L}$, whose vectors have size within a factor $\sqrt{\gamma/n}$ of the size of the smallest vector of $\mathcal{L}$ where $n = \dim(\mathcal{L})$, then the output of Step 3 is a solution to the approximate CVP with approximation factor γ (see [17, § 5] for a proof of that statement). According to Heuristic 2, $\operatorname{dist}(\vec{m}, \mathcal{L})$ is in $e^{(1+o(1))\log^{1/3}|D|}$. The approximation factor for the solution to γ-CVP of Step 3 satisfies $\gamma = nr^{2k/r} \ll L_D(1/3, 1 + o(1))$, and hence the output of Algorithm 7 is a vector with coefficients in $L_D(1/3, 1 + o(1))$. □

Comparison of the theoretical complexity with other methods. In [3, 11], another decomposition method was used. Let us compare the run-time of Algorithm 7 with the algorithm of [3, 11]. Our algorithm decomposes an input ideal $\mathfrak{a}$ in $\mathrm{Cl}(\mathcal{O})$ with respect to prime ideals of norm bounded by B, where B is chosen by the user. It also produces random relations in $\mathrm{Cl}(\mathcal{O})$ between primes of norm less than a bound B (by setting $\mathfrak{a} = \mathcal{O}$). For the applications to isogeny evaluation and endomorphism ring computation, we are interested in choosing a small B. In [3, 11], relations with smoothness bound $L_\Delta(1/2, 1/2\sqrt{3})$ are derived. The complexity of the method used in [3, 11] is in

$$\log(\mathcal{N}(\mathfrak{a}))^{1+o(1)} + L_D(1/2, \sqrt{3}/2 + o(1)) \quad \text{where } D = \operatorname{disc}(\mathcal{O}).$$

For the same smoothness bound B, our method for deriving relations in $\mathrm{Cl}(\mathcal{O})$ offers an asymptotic speed-up in two typical scenarios:

- when the ideal class group of $\mathrm{Cl}(\mathcal{O}_K)$ is known in advance; and
- when $\mathcal{O} \subsetneq \mathcal{O}_K$.

In the event that $\mathcal{O} = \mathcal{O}_K$ and that $\mathrm{Cl}(\mathcal{O}_K)$ is not known, our method still provides an asymptotic speed-up if we seek short relations with a smaller smoothness bound B. However, if one only seeks one random relation between primes of norm bounded by $L_\Delta(1/2, 1/2\sqrt{3})$, then the complexity of the computation of $\mathrm{Cl}(\mathcal{O}_K)$ will dominate the run-time. This is expected as there is no reason why the computation of one random relation without any particular restriction would be any slower than the computation of the entire lattice of relations. In [11], it was shown that these relations were asymptotically optimal for isogeny evaluation. In practice, however, we observed that evaluating the action of large primes comes at a high cost (see § 7). Deriving short relations for a small smoothness bound B is therefore very relevant to isogeny evaluation and we can easily show that, even when $\mathcal{O} = \mathcal{O}_K$ and when $\mathrm{Cl}(\mathcal{O}_K)$ is unknown, our algorithm performs asymptotically better than the methods of [3, 11] when looking for relations involving primes of norm bounded by $B \leqslant L_D(1/2, \sqrt{2}/6)$.

https://doi.org/10.1112/S1461157016000358 Published online by Cambridge University Press

6.2. *Comparative timings*

Timings for a decomposition in $\mathrm{Cl}(\mathcal{O}_K)$. We present numerical experiments highlighting the impact of our decomposition method when working in a maximal order of large discriminant. We compared our method to Sutherland's SmoothRelation C code which was used in [3] for computing endomorphism rings and in [11] for evaluating isogenies. We considered quadratic orders of (fundamental) discriminant $\Delta = -(10^{30}+57)$, $\Delta = -(10^{40}+121)$, $\Delta = -(10^{45}+9)$ and $\Delta = -(10^{50}+151)$. For each discriminant, we decomposed one of the prime ideals above the first twenty Elkies primes above $|\Delta|/2$ over the split primes of norm less than B where B ranges between 100 and 500. We compared the average time over the twenty instances with the amortized time, which assumes that the precomputation of the ideal class group of $\mathcal{O}_K$ is done only once. Besides the timings, we also reported an average score of the relations that were found. This score consists of the theoretical complexity of the evaluation of the action of a relation of the form $\mathfrak{a} \sim \prod_i \mathfrak{p}_i^{e_i}$ on an elliptic curve with endomorphism ring $\mathcal{O}_K$. According to [11], this cost is proportional to $\sum_i p_i^{3+\varepsilon} + |e_i| p_i^{2+\varepsilon}$ for $\varepsilon > 0$ arbitrarily small and where $p_i = \mathcal{N}(\mathfrak{p}_i)$. A high score means that the relation is of bad quality because the estimated time to evaluate the corresponding isogeny is high. The results are presented in Table A.1, in Appendix A. Our method achieves a significant speed-up for non-trivial instances. For example, it is about 100 times faster when $B = 260$ and allows us to reach a lower smoothness bound than SmoothRelation at no extra cost.

Timings for a decomposition in $\mathrm{Cl}(\mathcal{O})$ *when* $\mathcal{O} \subsetneq \mathcal{O}_K$. We now illustrate the impact of deriving relations in $\mathrm{Cl}(\mathcal{O})$ from relations in $\mathrm{Cl}(\mathcal{O}_K)$. We chose a series of quadratic orders $\mathcal{O}_4 \subset \mathcal{O}_3 \subset \mathcal{O}_2 \subset \mathcal{O}_1 \subset \mathcal{O}_K$, where $\mathcal{O}_K$ is the ring of integers of $K = \mathbb{Q}(\sqrt{-7})$. This corresponds to orders illustrating Bisson and Sutherland's algorithm for computing the $\mathrm{End}(E)$ [3, § 5.2]. The discriminants of the quadratic orders are:

- $\mathrm{disc}(\mathcal{O}_1) = -7 \cdot 852857^2$;
- $\mathrm{disc}(\mathcal{O}_2) = -7 \cdot 852857^2 \cdot 582511^2$;
- $\mathrm{disc}(\mathcal{O}_3) = -7 \cdot 852857^2 \cdot 582511^2 \cdot 582509^2$; and
- $\mathrm{disc}(\mathcal{O}_4) = -7 \cdot 852857^2 \cdot 582511^2 \cdot 582509^2 \cdot 2305843009213693951^2$.

We followed the same protocol as in the previous paragraph. For each order $\mathcal{O}_i$, where $i = 1, \ldots, 4$, we decomposed the first twenty Elkies primes over $|D_i|/2$ in $\mathrm{Cl}(\mathcal{O}_i)$, where $D_i = \mathrm{disc}(\mathcal{O}_i)$. We repeated the experiment for different values of the smoothness bound. The timings are presented in Table B.1, in Appendix B. We observed a significant speed-up for non-trivial inputs ($\mathcal{O}_3$ and $\mathcal{O}_4$). For example, the average time to find a decomposition in $\mathrm{Cl}(\mathcal{O}_3)$ with a smoothness bound of $B = 200$ with our method is 0.12 s, while it is 133.06 s with SmoothRelation for a comparable score. As before, our method does not suffer from restrictions on the size of the factor base. In this case, as the ideal class group of $\mathcal{O}_K$ is particularly small, it is, in fact, faster to derive shorter relations.

7. *Isogeny evaluation*

We show the impact of § 5 on the evaluation of an isogeny. Let E be an ordinary elliptic curve defined over $\mathbb{F}_p$ with trace t. The endomorphism ring $\mathrm{End}(E)$ is isomorphic to an order $\mathcal{O}$ inside $K = \mathbb{Q}(\sqrt{t^2 - 4p})$. Given a prime ideal $\mathfrak{L}$ of K of large norm, we show how to evaluate, up to isomorphisms of E, the unique normalized horizontal isogeny whose kernel is isomorphic to $\mathfrak{L} \cap \mathcal{O}$. Our approach follows [11], which derives from [4].

7.1. *Description of the algorithm*

Algorithm 8 describes how to evaluate a large-degree isogeny at a point on an ordinary curve using an ideal decomposition method. This corresponds to [4, Algorithm 4.1] and [11, Algorithm 1].

https://doi.org/10.1112/S1461157016000358 Published online by Cambridge University Press

ALGORITHM 8. Isogeny evaluation.

```
Require: $E$ defined over $\mathbb{F}_q$ of characteristic $p$, $P \in E(\mathbb{F}_{p^n})$ with $\gcd([\mathcal{O}_K : \mathbb{Z}[\pi], \#E(\mathbb{F}_{p^n}) = 1)$ and $\mathfrak{L} \subseteq \mathcal{O}_K$ of prime norm $\ell \neq p \nmid [\mathcal{O}_K : \mathbb{Z}[\pi]]$.
Ensure: $E'$ admitting a normalized isogeny $\phi : E \to E'$ with kernel $E[\mathfrak{L}]$ and $x(\phi(P))$.
Find $\mathcal{O} \simeq \mathrm{End}(E)$ with Algorithm 9.
Decompose $\mathfrak{L} \cap \mathcal{O} \sim \prod_i \mathfrak{p}_i^{e_i}$ with $|e_i| \in L_D(1/3, 1 + o(1))$ with Algorithm 7.
Find $\alpha$ such that $\mathfrak{L} \cap \mathcal{O} = (\alpha) \prod_i \mathfrak{p}_i^{e_i}$ using Cornacchia's algorithm.
Repeat the method of [4, § 3.1] to find $\phi_c : E \to E_c$ with kernel $E[\prod_i \mathfrak{p}_i^{e_i}]$.
Evaluate $\phi_c(P) \in E_c[\mathbb{F}_{q^n}]$.
Write $\alpha = (u + v\pi)/(zm)$. Find $\eta : E_c \to E'$ with $\eta^*(\omega_{E'}) = (u/zm)\omega_{E_c}$, using [4, § 4].
Compute $Q = \eta(\phi_c(P))$ and $(zm)^{-1} \bmod \#(E(\mathbb{F}_{q^n}))$. $R \leftarrow ((zm)^{-1}(u + v\pi))(Q)$.
return $E', x(R)$.
```

PROPOSITION 7.1 (GRH) + (NFS) + (H). *Let $C > 0$ be a constant. Suppose that* $\mathrm{End}(E) \simeq \mathcal{O} \in \mathcal{Q}_C$ *with* $\mathrm{disc}(\mathcal{O}) = D$ *and let f be the conductor of $\mathbb{Z}[\pi]$, where π is the Frobenius endomorphism and $\mathcal{O}_K$ is the maximal order containing $\mathcal{O}$. The complexity of Algorithm 8 is in*

$$\log(\ell)^{1+o(1)} + L_f(1/3, \sqrt[3]{64/9} + o(1)) + L_D(1/3, 1 + o(1)) + L_\Delta(1/2, 3/\sqrt{8} + o(1)),$$

with $\Delta = \mathrm{disc}(\mathcal{O}_K)$. *Moreover, the run-time of subsequent isogeny computation* (*for any curve E' with* $\mathrm{End}(E') \simeq \mathcal{O}' \subseteq \mathcal{O}_K$) *is in* $\log(\ell)^{1+o(1)} + L_v(1/3, \sqrt[3]{64/9} + o(1)) + L_D(1/3, 1 + o(1)) + L_\Delta(1/2, \sqrt{2}/2 + o(1))$. *This complexity also holds true for the first isogeny evaluation if* $\mathrm{Cl}(\mathcal{O}_K)$ *is known in advance.*

The algorithm of Jao and Soukharev runs in time $\log(\ell)^{1+o(1)} + L_f(1/3, \sqrt[3]{64/9} + o(1)) + L_{D_\pi}(1/2, \sqrt{3}/2 + o(1))$ if we incorporate the computation of $\mathrm{End}(E)$ in the evaluation of ϕ. If not, the complexity of Jao and Soukharev's method becomes $\log(\ell)^{1+o(1)} + L_D(1/2, \sqrt{3}/2 + o(1))$. Therefore, our method provides an asymptotic speed-up when $\mathcal{O} \subsetneq \mathcal{O}_K$ and when $\mathrm{Cl}(\mathcal{O}_K)$ is known in advance. In addition, as we see in § 7.2, we obtain a significant speed-up in practice even when $\mathcal{O} = \mathcal{O}_K$ and $\mathrm{Cl}(\mathcal{O}_K)$ is not known because of the cost of evaluating the action of large primes of large size (as discussed in § 7.2).

7.2. *Numerical experiments*

The cost of the action of $\mathfrak{p}$ in practice. Our numerical experiments highlighted the cost of the evaluation of the action of a prime $\mathfrak{p}$ on the curve E when $\mathcal{N}(\mathfrak{p})$ is large (above 300) and when the characteristic of the field of definition of E is large. This means that even when our method cannot leverage a precomputation of $\mathrm{Cl}(\mathcal{O}_K)$ or take advantage of a large gap between $\mathcal{O}$ and $\mathcal{O}_K$, we still obtain a significant speed-up when working with non-trivial examples. Indeed, our algorithm returns very short relations that only involve primes whose action is easy to evaluate. According to [11], the cost of evaluating a relation is dominated by the computation of the $\Phi_\ell(X, Y) \bmod p$, where ℓ is the norm of prime ideals occurring in the relation. The computation of $\Phi_\ell(X, Y) \bmod p$ can be expensive, in practice, as shown by [5, Table 3]. Other modular polynomials can be used on restricted classes of curves, for example the Weber polynomials. Also, the methods of [18] allow us to compute $\Phi_\ell(X, j(E))$, given $j(E)$ and ℓ, significantly faster than the computation of Φ_ℓ. However, our numerical experiments highlighted the high cost of computing the action of $\mathfrak{p}$ given Φ_ℓ, where $\ell = \mathcal{N}(\mathfrak{p})$. We collected the run-time of the execution of the Atkin–Elkies technique for split primes $\ell < 300$ using the modular polynomials available on Sutherland's web page on the curve $E : y^2 = x^3 + ax + b$

https://doi.org/10.1112/S1461157016000358 Published online by Cambridge University Press

over $\mathbb{F}_p$, where

$$\begin{aligned} p &= 862591559561497151050143615844796924047865589835498401307522524859467869 \\ a &= 201251174924006028393812367563624537259760372830791045273179137590736622 \\ b &= 545482459632327583111433582031095022426858572446976004219654298705912499. \end{aligned}$$

Table C.1 shows that the cost of running the Atikin–Elkies technique is, in practice, higher than the cost of computing the modular polynomials. To the best of our knowledge, there is no record in the literature of the execution of the Atkin–Elkies technique for non-trivial values of $\mathfrak{p}$. Indeed, the computations of [4, 11] were made on the precomputed modular polynomials available in Magma of lever less than 60. This makes our short relation algorithm even more relevant in practice than what theory suggests.

ECC_p239: a large example. Our first example to illustrate the impact of our ideal decomposition technique on isogeny evaluation is the curve ECC_p239 of the Certicom challenge. As in [11, § 5.3], we decompose the first Elkies prime above $p/2$. In [11, § 5.3], Jao and Soukharev only decomposed the input ideal over primes of norm less than 5000 using SmoothRelation. Over twenty random instances, the average time on a Intel Core i7-2600 CPU at 3.40 GHz to find a smooth relation was 21.5 h, and the estimated time to compute the corresponding modular polynomials was 12.3 h. Our method returned a short decomposition $\mathfrak{L} \sim \mathfrak{p}_{277}^{4}\mathfrak{p}_{271}^{-3}\mathfrak{p}_{269}^{1}\mathfrak{p}_{257}^{5}\mathfrak{p}_{239}^{1}\mathfrak{p}_{211}^{-3}\mathfrak{p}_{199}^{-6}\mathfrak{p}_{197}^{-1}\mathfrak{p}_{193}^{-5}\mathfrak{p}_{179}^{4}\mathfrak{p}_{167}^{-3}\mathfrak{p}_{163}^{10}\mathfrak{p}_{151}^{-7}\mathfrak{p}_{137}^{8}\mathfrak{p}_{131}^{8}\mathfrak{p}_{113}^{-2}\mathfrak{p}_{97}^{5}\mathfrak{p}_{89}^{1}\mathfrak{p}_{83}^{-3}\mathfrak{p}_{71}^{2}\mathfrak{p}_{59}^{-3}\mathfrak{p}_{37}^{1}\mathfrak{p}_{29}^{-2}$ $\mathfrak{p}_{23}^{-2}\mathfrak{p}_{19}^{-3}\mathfrak{p}_{11}^{-8}\mathfrak{p}_{7}^{3}$ in 2.7 h, which we were able to evaluate at the point P of the challenge in 47 min. The time to find subsequent relations is under 100 s.

ECC_p359: a very large example. We handled this example which was out of the reach of the methods developed in [11]. We evaluated an isogeny of degree ℓ, where ℓ is the next Elkies prime above $p/2$ on the curve ECC_p359 from the Certicom challenge. Our method returned the short decomposition $\mathfrak{L} \sim \mathfrak{p}_{293}^{-22}\mathfrak{p}_{277}^{15}\mathfrak{p}_{263}^{4}\mathfrak{p}_{251}^{8}\mathfrak{p}_{239}^{17}\mathfrak{p}_{229}^{19}\mathfrak{p}_{227}\mathfrak{p}_{223}^{19}\mathfrak{p}_{211}^{-15}\mathfrak{p}_{191}^{-10}\mathfrak{p}_{179}^{-11}\mathfrak{p}_{173}^{-12}\mathfrak{p}_{157}^{-15}\mathfrak{p}_{151}\mathfrak{p}_{137}^{-2}$ $\mathfrak{p}_{131}^{-1}\mathfrak{p}_{113}^{18}\mathfrak{p}_{103}^{-10}\mathfrak{p}_{101}^{26}\mathfrak{p}_{71}^{-10}\mathfrak{p}_{67}^{-12}\mathfrak{p}_{61}^{13}\mathfrak{p}_{59}^{20}\mathfrak{p}_{53}^{23}\mathfrak{p}_{37}^{-2}\mathfrak{p}_{31}^{15}\mathfrak{p}_{19}\mathfrak{p}_{17}^{-21}\mathfrak{p}_{7}^{-7}$. We computed the relation matrix in six days on 64 cores Intel X7560 Xeon at 2.27 GHz. Computing its HNF took an extra eight days on 64 cores using Gaussian elimination and the HNF algorithm of [15]. The first factorization of the ideal took four days on one core, and the search for a small relation took only two minutes. The subsequent evaluation of the isogeny corresponding to $\mathfrak{L}$ at the point P of the challenge took five hours. The bottleneck of the evaluation is because the repeated execution of the Atkin–Elkies, using Sutherland's precomputed modular polynomials of level up to 300, leads to the curve $E : y^2 = x^3 + a_c x + b_c$, where

$$\begin{aligned} a_c &= 0x17926806106C0B651E621375531E008FFA57A529DF58B2CB4BAE28 \\ &\quad 794301EC671638134938F6BF8C8110539B98, \\ b_c &= 0x3CD8EC661E34C138DC8619B776B4464194D22A53797E5360A81AC64 \\ &\quad 16E27A50634F7CD57765113325DBBF845AD. \end{aligned}$$

This took 17954.42 CPU sec on an Intel Core i7-2600 CPU at 3.40 GHz with 8 GB of memory. The image of P by the corresponding isogeny ϕ_c is

$$\begin{aligned} P_c = (&0x32E0033B0ECFDE10D1D4A3924267356E71C05C62A3F47C0408457B \\ &E6C150A44540F377277B7214CF9D55DEE14F \\ &: 0x61C3257F04B967F180FA47237A73B7660158E344959BB982BACC9F0 \\ &9843260F4079AC0B69E840E602CFCAEB3D : 1). \end{aligned}$$

https://doi.org/10.1112/S1461157016000358 Published online by Cambridge University Press

Using the method of [4, § 4], we find $\eta : E \to E'$ and $E' : y^2 = x^3 + a'x + b'$ over $\mathbb{F}_p$, where

$$\begin{aligned} a' &= 0x3C11F6F60BE244C7F3CA424353FB5AA6E307ECBCB1FD46523A64EC \\ &\quad C66DCBA2650CA3F5F86AE07FDA4ECB70DECB, \\ b' &= 0x1348E3530180FA7FF30F387E52ACFB1BE7945AB2F910C63891E4D06 \\ &\quad 14F0071FFE13C57B787DC0D590456350E39. \end{aligned}$$

Then we compute u, v, z, m such that $\alpha = (u + v\pi)/(zm)$, $Q = \eta(P_c)$ and $P' = ((zm)^{-1} (u + v))Q$.

$$\begin{aligned} P' = (&0x35CAE02BC65146B0252B5CE0FDEBA04205AE070128993BB208A7E2 \\ &5DE06404D2F2CCFAB21EBB3 \\ &: 0x40DAAA91EABF3D3B6ED5368E9A667289C6CA72CE89DE24046CF282 \\ &EF7AA1E18CB50F8EBFAEDCDB4F50B96A07D : 1). \end{aligned}$$

8. *Calculating* End(E)

We also successfully applied our ideal decomposition technique to the computation of the endomorphism ring of an ordinary elliptic curve E over $\mathbb{F}_p$. We followed the approach of Bisson and Sutherland. Given the maximal order $\mathcal{O}_K$ and $\mathbb{Z}[\pi]$, where π is the Frobenius endomorphism and $K = \mathbb{Q}(\sqrt{t^2 - 4p})$, we compute the order $\mathcal{O}$ which is isomorphic to $\mathrm{End}(E)$. The orders $\mathbb{Z}[\pi] \subseteq \mathcal{O} \subseteq \mathcal{O}_K$ are identified by their conductor u, which satisfies $1 \leqslant u \leqslant f$, where f is the conductor of $\mathbb{Z}[\pi]$.

8.1. *Theoretical complexity of computing* End(E)

The result of Bisson and Sutherland consists of two different methods for computing u. From below: for increasingly large divisors u of f, try to certify with [3, Alg. Certify] if the conductor of $\mathrm{End}(E)$ is u. From above: for each $p \mid f$, test if $p \mid u$ (corresponds to Algorithm 9. Both strategies rely on two ingredients: finding random relations in $\mathrm{Cl}(\mathcal{O})$ for some $\mathbb{Z}[\pi] \subseteq \mathcal{O} \subseteq \mathcal{O}_K$ and testing if the action of a product $\prod_i \mathfrak{p}_i^{e_i}$ on the isomorphism class of E is trivial. Therefore, substituting the random walk relation search in $\mathrm{Cl}(\mathcal{O})$ by the methods we described in § 5 provides a significant speed-up.

Algorithm 9. Computing End(E) (from above).

```
Require: $E$ of trace $t$ defined over $\mathbb{F}_p$, $D_K = \mathrm{disc}(\mathcal{O}_K)$ and $v$ conductor of $\mathbb{Z}[\pi]$ in $\mathcal{O}_K$.
Ensure: $u$ such that $\mathrm{End}(E)$ is the quadratic order of order $u$.
$u \leftarrow 1$.
for $p \mid f$ do
    for $k \leqslant v_p(f)$ do
        Let $\mathcal{O}_1$ with $\mathrm{disc}(\mathcal{O}_1) = (f/p^j)^2 D_K$ and $\mathcal{O}_2$ with $\mathrm{disc}(\mathcal{O}_2) = p^{2k} D_K$.
        Find a relation $\mathcal{R}$ holding in $\mathrm{Cl}(\mathcal{O}_1)$ but not in $\mathrm{Cl}(\mathcal{O}_2)$. We denote it by $\mathcal{R}/\mathcal{O}_1$.
        If $\mathcal{R}/\mathcal{O}_1$ does not hold in $\mathrm{Cl}(\mathrm{End}(E))$: $u \leftarrow u \cdot p$.
    end for
end for
return $u$.
```

https://doi.org/10.1112/S1461157016000358 Published online by Cambridge University Press

Proposition 8.1 (GRH) + (NFS) + (H). *Let $C > 0$ be a constant. Let f be the conductor of $\mathbb{Z}[\pi]$, let $D_\pi = \mathrm{disc}(\mathbb{Z}[\pi])$ and let $\Delta = \mathrm{disc}(\mathcal{O}_K)$, where $\mathcal{O}_K \in \mathcal{Q}_C$ is the maximal order containing $\mathbb{Z}[\pi]$. The complexity of computing* End(E) *with Algorithm* 9 (*from above*), *using the short relation generation methods described in* § 5 *is in*

$$L_f(1/3, \sqrt[3]{64/9} + o(1)) + L_{D_\pi}(1/3, 1 + o(1)) + L_\Delta(1/2, 3/\sqrt{8} + o(1)).$$

The endomorphism ring computation ‘from below’ is described in detail in [3, Algorithm 2]. Let $D = \mathrm{disc}(\mathrm{End}(E))$. The complexity of the methods described in [3] is:

- $L_f(1/3, \sqrt[3]{64/9} + o(1)) + L_{D_\pi}(1/2, \sqrt{3}/2 + o(1))$ when computing End(E) from above; and
- $L_f(1/3, \sqrt[3]{64/9} + o(1)) + L_D(1/2 + \varepsilon, 1)$ for arbitrarily small $\varepsilon > 0$ from below.

Our method is asymptotically faster in all cases except when $\mathbb{Z}[\pi] = \mathcal{O}_K$, but, in this case, the answer $\mathrm{End}(E) = \mathcal{O}_K$ to our problem is trivial.

8.2. *Numerical experiments*

Finding interesting examples for the problem of computing End(E) is a hard problem. To illustrate the speed-up provided by our ideal decomposition technique, we computed End(E) from above, where E is the example provided by Bisson and Sutherland [3, § 5.2]. We ran this computation only for comparison purposes, as the approach from below works better for this particular curve. The conductor v of $\mathbb{Z}[\pi]$ factors as

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

The primes 2 and 127 are treated by isogeny climbing. Then we perform the following tests.

- $p_1 \nmid u$: we get $\mathcal{R}/\mathcal{O}_1$ in 0.06 s and check it does not hold in Cl(End(E)) in 465 s.
- $p_2 \mid u$: we get $\mathcal{R}/\mathcal{O}_2$ in 0.07 s and check it holds in Cl(End(E)) in 540 s.
- $p_3 \nmid u$: we get $\mathcal{R}/\mathcal{O}_3$ in 0.07 s and check it holds in Cl(End(E)) in 426 s.
- $p_4 \nmid u$: we get $\mathcal{R}/\mathcal{O}_4$ in 0.03 s and check it holds in Cl(End(E)) in 108 s.

In comparison with relations derived with SmoothRelation, the computation of End(E) from above takes the better part of a day [3, § 5.2]. Indeed, according to a private communication of Sutherland, the total time to produce B-smooth relations with $B = 4000$ for all orders is fifteen minutes. Then, as Weber polynomials do not apply to this particular curve, the evaluation of the action of each prime of norm more than 2000 takes at least an hour. The exact relations $\mathcal{R}/\mathcal{O}_i$ can be found in Appendix C.

9. *Conclusion and further perspectives*

Curves of higher genus. Our ideal decomposition technique readily applies to orders in families of number fields with fixed degree. Therefore, it can be used for evaluating isogenies and computing the endomorphism ring of abelian varieties with complex multiplication by such orders. In particular, we anticipate that it will be rather straightforward to implement our techniques for the genus two case since we can substitute the j invariants and the corresponding modular polynomials by their generalization [8].

Quantum algorithms. Our ideal decomposition algorithm minimizes the impact of the execution of the BKZ algorithm. Quantum computers can compute Cl($\mathcal{O}$) and factor the conductor of $\mathcal{O}$ in polynomial time. Therefore, the run- time of the decomposition of $\mathfrak{a} \subseteq \mathcal{O}$ is dominated by the BKZ algorithm (in $L_D(1/3, 1 + o(1))$). Applying our method to isogeny evaluation and to the computation of End(E), yields a quantum $L_D(1/3, 1 + o(1))$ complexity.

https://doi.org/10.1112/S1461157016000358 Published online by Cambridge University Press

Proposition 9.1. *Given an ordinary elliptic curve E over $\mathbb{F}_q$ and $\ell > 0$, there is a heuristic quantum algorithm for computing* $\mathrm{End}(E)$ *in time* $L_D(1/3, 1 + o(1))$ *and for evaluating a horizontal l-isogeny on E in time* $L_D(1/3, 1 + o(1))$*, where* $D = \mathrm{disc}(\mathrm{End}(E))$ *and where the polynomial factors are omitted.*

The previous state-of-the-art was $L_D(1/2, O(1))$. Fast evaluation of the action of an ideal is a key ingredient of the subexponential quantum algorithm of Jao and Soukharev to compute an isogeny between two input curves [6]. Sadly, our method does not provide a better run-time for this task because Kuperberg's sieve, which was used to solve the isogeny problem, requires $L_D(1/2, O(1))$ calls to an oracle evaluating the action of an ideal, and thus dominates the run-time.

Supersingular curves. Given an order $\mathcal{O}$ in a quaternion algebra, the ℓ-isogeny path in $\mathcal{O}$ was recently shown to be easy [13]. Therefore, the computation of the endomorphism ring of a supersingular curve is very important for isogeny computation. Whether the search for small decompositions of ideals in orders could be leveraged to identify the endomorphism ring of a supersingular curve remains an open problem to this date.

Appendix A. *Short decompositions in* $\mathrm{Cl}(\mathcal{O}_K)$

Timings are quoted in CPU seconds. They were obtained on a Intel Core i7-2600 CPU at 3.40 GHz with 8 GB of memory. Timings for the SmoothRelation correspond to v1.3 compiled with GMP 6.1.0. Our methods were implemented with Magma v.2.21-7, which includes Biasse's implementation of Jacobson's quadratic sieve algorithm for computing the class group. Empty spaces correspond to input values for which SmoothRelation could not complete the computational tasks. The amortized time correspond to the average time over twenty instances, where the ideal class group is computed only once.

Table A.1. *Decomposition of the first* 20 *Elkies primes above* $|D|/2$ *in* $\mathrm{Cl}(\mathcal{O}_K)$.

Data		This paper			SmoothRelation	
Δ	B	Average t	Amortized t	Average score	Average t	Average score
	300	4.65	0.41	126 715 962	0.39	44 697 833
	280	4.64	0.40	134 701 855	0.58	42 440 149
	260	4.63	0.39	98 368 419	1.65	32 718 214
	240	4.63	0.39	59 695 240	1.89	28 583 231
	220	4.62	0.39	40 098 146	3.03	11 109 250
$-(10^{30} + 57)$	200	4.62	0.38	40 098 146	5.15	10 574 549
	180	4.62	0.38	15 170 525	233.39	6 587 674
	160	4.62	0.38	15 170 525	101.64	7 649 859
	140	4.62	0.39	8 783 898	147.60	5 171 428
	120	4.62	0.39	5 796 284		
	100	4.62	0.39	3 611 129		

https://doi.org/10.1112/S1461157016000358 Published online by Cambridge University Press

Table A.1. (*Continued*).

Data		This paper			SmoothRelation	
Δ	B	Average t	Amortized t	Average score	Average t	Average score
$-(10^{40}+121)$	400	6.73	0.73	262 077 045	12.70	60 255 753
	300	6.75	0.75	76 019 864	20.62	30 047 430
	280	6.67	0.67	76 019 864	12.54	23 127 191
	260	6.67	0.67	106 075 607	29.33	24 885 172
	240	6.65	0.65	71 754 968	32.94	22 797 040
	220	6.65	0.64	41 494 637	865.83	15 941 400
	200	6.64	0.63	35 837 955	1110.38	10 165 036
	180	6.64	0.64	13 966 941		
	160	6.65	0.64	10 231 198		
	140	6.64	0.63	10 231 198		
	120	6.68	0.67	5 423 093		
	100	6.77	0.76	5 423 093		
$-(10^{45}+9)$	500	32.88	7.36	921 100 376	281.01	250 827 784
	400	27.54	2.03	572 851 862	523.94	129 440 986
	300	27.39	1.87	178 659 605	4938.13	38 651 234
	280	27.39	1.87	85 642 798	3344.17	32 955 470
	260	27.35	1.83	87 776 993	2120.11	32 955 470
	240	27.36	1.84	79 420 010		
	220	27.35	1.83	43 295 409		
	200	27.39	1.87	43 295 409		
	180	27.47	1.95	23 334 616		
	160	27.71	2.19	15 065 406		
	140	27.54	2.03	10 655 678		
	120	27.64	2.12	7 742 969		
	100	27.58	2.06	4 000 636		
$-(10^{50}+151)$	500	248.03	219.63	579 286 470	538.46	169 469 117
	400	37.98	9.57	502 925 202	916.25	84 257 446
	300	31.19	2.79	218 772 201	1232.27	60 835 899
	280	31.00	2.59	131 830 056	8039.38	25 531 505
	260	30.95	2.55	80 140 272		
	240	30.88	2.48	73 607 358		
	220	30.99	2.58	59 357 173		
	200	30.84	2.43	51 336 782		
	180	30.81	2.40	25 266 722		
	160	30.91	2.51	18 419 405		
	140	30.88	2.47	12 803 933		
	120	30.86	2.45	7 280 943		
	100	30.81	2.41	3 536 980		

Appendix B. *Short decompositions in* $\mathrm{Cl}(\mathcal{O})$

Timings t are quoted in CPU seconds. They were obtained on a Intel Core i7-2600 CPU at 3.40 GHz with 8 GB of memory. Timings for the SmoothRelation correspond to v1.3 compiled with GMP 6.1.0. Our methods were implemented with Magma v.2.21-7, which includes Biasse's implementation of Jacobson's quadratic sieve algorithm for computing the class group. Empty spaces correspond to input values for which SmoothRelation could not complete the computational tasks. The amortized time correspond to the average time over twenty instances, where the ideal class group is computed only once.

TABLE B.1. *Decomposition of the first* 20 *Elkies primes above* $|D|/2$ *in* $\mathrm{Cl}(\mathcal{O})$.

Data		This paper			SmoothRelation	
Δ	B	Average t	Amortized t	Average score	Average t	Average score
$\mathcal{O}_1$	400	0.12	0.10	166 567 858	0.01	44 049 959
	350	0.08	0.06	95 532 880	0.01	35 514 353
	300	0.07	0.05	24 732 420	0.01	21 556 463
	250	0.06	0.04	35 402 798	0.01	12 927 271
	200	0.05	0.03	13 006 394	0.01	8 360 212
	150	0.04	0.02	8 621 110	0.01	2 165 192
$\mathcal{O}_2$	400	0.29	0.28	313 425 886	0.03	69 068 937
	350	0.12	0.11	104 228 139	0.03	48 508 682
	300	0.10	0.09	75 085 703	0.02	26 263 081
	250	0.12	0.11	64 896 476	0.03	17 048 229
	200	0.10	0.10	38 731 894	0.03	12 023 808
	150	0.07	0.06	6 985 717	0.06	4 582 418
$\mathcal{O}_3$	400	0.28	0.26	417 709 168	3.76	81 206 130
	350	0.20	0.18	202 624 787	8.70	63 054 195
	300	0.16	0.13	134 224 416	6.37	39 600 945
	250	0.14	0.11	53 268 050	9.51	22 026 158
	200	0.12	0.09	37 927 113	133.06	25 290 274
	150	0.09	0.06	12 822 693		
$\mathcal{O}_4$	400	0.42	0.33	455 234 165		
	350	0.35	0.26	245 653 258		
	300	0.22	0.13	145 753 178		
	250	0.20	0.10	75 975 948		
	200	0.19	0.09	52 021 907		
	150	0.16	0.07	20 857 964		

Appendix C. *Supplement on the isogeny numerical evaluations*

C.1. *The cost of the action of* $\mathfrak{p}$ *in practice*

The numerical results presented in Table C.1 obtained on a Intel Core i7-2600 CPU at 3.40 GHz with 8 GB of memory show a significantly slower run-time than the computation of the modular polynomials [5, Table 3] (which are themselves harder to compute than Weber polynomials and instantiated modular polynomials).

C.2. *Computation of* End(E) (§ 8)

We obtained our timings on a Intel Core i7-2600 CPU at 3.40 GHz with 8 GB of memory. We used the precomputed modular polynomials of level up to 300 available on Sutherland's web page. The relations mentioned in the numerical experiment are the following.

$$\mathcal{R}/\mathcal{O}_1 : \mathfrak{p}_{11}^{-8}\mathfrak{p}_{23}^{-7}\mathfrak{p}_{29}^{-10}\mathfrak{p}_{37}^{9}\mathfrak{p}_{43}^{-17}\mathfrak{p}_{53}^{-2}\mathfrak{p}_{67}^{11}\mathfrak{p}_{71}^{14}\mathfrak{p}_{79}^{-11}\mathfrak{p}_{107}^{-1}\mathfrak{p}_{109}^{10}\mathfrak{p}_{113}^{8}\mathfrak{p}_{137}^{-13}\mathfrak{p}_{149}^{4}\mathfrak{p}_{151}^{1}\mathfrak{p}_{163}^{-1}\mathfrak{p}_{179}^{4}\mathfrak{p}_{191}^{4}\mathfrak{p}_{193}^{11}\mathfrak{p}_{211}^{16} \sim (1)$$

$$\mathcal{R}/\mathcal{O}_2 : \mathfrak{p}_{11}^{2}\mathfrak{p}_{23}^{15}\mathfrak{p}_{29}^{-6}\mathfrak{p}_{37}^{-6}\mathfrak{p}_{43}^{9}\mathfrak{p}_{53}^{-10}\mathfrak{p}_{67}^{11}\mathfrak{p}_{71}^{5}\mathfrak{p}_{79}^{-2}\mathfrak{p}_{107}^{9}\mathfrak{p}_{109}^{-4}\mathfrak{p}_{113}^{9}\mathfrak{p}_{137}^{-3}\mathfrak{p}_{149}^{-12}\mathfrak{p}_{151}^{3}\mathfrak{p}_{163}^{9}\mathfrak{p}_{179}^{16}\mathfrak{p}_{191}^{4}\mathfrak{p}_{193}^{-9}\mathfrak{p}_{197}^{-13}\mathfrak{p}_{211}^{-1} \sim (1)$$

$$\mathcal{R}/\mathcal{O}_3 : \mathfrak{p}_{11}^{4}\mathfrak{p}_{23}^{-2}\mathfrak{p}_{29}^{-3}\mathfrak{p}_{37}^{4}\mathfrak{p}_{43}^{-1}\mathfrak{p}_{53}^{-7}\mathfrak{p}_{67}^{-7}\mathfrak{p}_{71}^{13}\mathfrak{p}_{79}^{-17}\mathfrak{p}_{107}^{-17}\mathfrak{p}_{113}^{-2}\mathfrak{p}_{137}^{5}\mathfrak{p}_{149}^{-22}\mathfrak{p}_{151}^{-8}\mathfrak{p}_{163}^{4}\mathfrak{p}_{179}^{-6}\mathfrak{p}_{191}^{-2}\mathfrak{p}_{193}^{-7}\mathfrak{p}_{211}^{-3} \sim (1)$$

$$\mathcal{R}/\mathcal{O}_4 : \mathfrak{p}_{11}^{46}\mathfrak{p}_{23}^{-29}\mathfrak{p}_{29}^{-24}\mathfrak{p}_{37}^{-47}\mathfrak{p}_{43}^{21}\mathfrak{p}_{53}^{-25}\mathfrak{p}_{67}^{-67}\mathfrak{p}_{71}^{42}\mathfrak{p}_{79}^{79} \sim (1).$$

TABLE C.1. *Cost of evaluating the action of* $\mathfrak{p}$.

$\mathcal{N}(\mathfrak{p})$	Run-time of the Atkin–Elkies method [4, § 3.1]
11	0.070
13	0.060
19	0.130
23	0.270
29	0.190
37	0.400
59	2.710
67	3.310
71	3.540
83	7.150
89	1.990
97	2.690
101	9.770
113	8.600
131	14.450
137	13.680
151	19.580
163	13.800
167	20.880
179	122.630
193	16.920
197	30.040
199	25.980
211	60.810
239	33.460
257	55.390
269	63.170
271	164.630
277	72.810
283	68.910

https://doi.org/10.1112/S1461157016000358 Published online by Cambridge University Press

References

1. M. Ajtai, R. Kumar and D. Sivakumar, 'A sieve algorithm for the shortest lattice vector problem', *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing, STOC '01* (ACM, New York, 2001) 601–610.

2. G. Bisson, 'Computing endomorphism rings of elliptic curves under the GRH', *J. Math. Cryptol.* 5 (2012) no. 2, 101–114.

3. G. Bisson and A. Sutherland, 'Computing the endomorphism ring of an ordinary elliptic curve over a finite field', *J. Number Theory* 131 (2011) no. 5, 815–831. Elliptic Curve Cryptography.

4. R. Bröker, D. Xavier Charles and K. Lauter, 'Evaluating large degree isogenies and applications to pairing based cryptography', *Pairing-based Cryptography – Pairing 2008, Proceedings of Second International Conference,* Egham, September 1–3, 2008, Lecture Notes in Computer Science 5209 (eds S. Galbraith and K. Paterson; Springer, 2008) 100–112.

5. R. Bröker, K. Lauter and D. Sutherland, 'Modular polynomials via isogeny volcanoes', *Math. Comput.* 81 (2012) 1201–1231.

6. A. Childs, D. Jao and V. Soukharev, 'Constructing elliptic curve isogenies in quantum subexponential time', *J. Math. Cryptol.* 8 (2013) no. 1, 1–29.

7. H. Cohen, *A course in computational algebraic number theory*, Graduate Texts in Mathematics 138 (Springer, 1991).

8. R. Dupont, 'Moyenne arithmético-géométrique, suites de Borchardt et applications', PhD Thesis, École Polytechnique, 2006.

9. J. L. Hafner and K. S. McCurley, 'A rigorous subexponential algorithm for computation of class groups', *J. Amer. Math. Soc.* 2 (1989) 839–850.

10. D. Jao, S. D. Miller and R. Venkatesan, 'Expander graphs based on GRH with an application to elliptic curve cryptography', *J. Number Theory* 129 (2009) no. 6, 1491–1504.

11. D. Jao and V. Soukharev, 'A subexponential algorithm for evaluating large degree isogenies', *Algorithmic number theory*, Lecture Notes in Computer Science 6197 (eds G. Hanrot, F. Morain and E. Thomé; Springer, Berlin, Heidelberg, 2010) 219–233.

12. J. Klüners and S. Pauli, 'Computing residue class rings and picard groups of orders', *J. Algebra* 292 (2005) no. 1, 47–64.

13. D. Kohel, K. Lauter, C. Petit and J.-P. Tignol, 'On the quaternion l-isogeny path problem', *LMS J. Comput. Math.* 17 (2014) 418–432; 1.

14. A. K. Lenstra, H. W. Lenstra Jr., M. S. Manasse and J. M. Pollard, 'The number field sieve', *STOC '90: Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing* (ACM, New York, 1990) 564–572.

15. C. Pauderis and A. Storjohann, 'Computing the invariant structure of integer matrices: fast algorithms into practice', *International Symposium on Symbolic and Algebraic Computation, ISSAC'13,* Boston, MA, June 26–29, 2013 (ed. M. Kauers; ACM, New York, 2013) 307–314.

16. R. Schoof, 'Counting points on elliptic curves over finite fields', *J. Théor. Nombres Bordeaux* 7 (1995) 219–254.

17. N. Stephens-Davidowitz, 'Dimension-preserving reductions between lattice problems', http://www.noahsd.com/latticeproblems.pdf.

18. A. Sutherland, 'On the evaluation of modular polynomials', *Proceedings of the Tenth Algorithmic Number Theory Symposium (ANTS X)*, Open Book Series 1 (Mathematical Sciences Publishers, Berkeley, CA, 2013) 531–555.

Jean-François Biasse
University of South Florida
USA

biasse@usf.edu

Claus Fieker
University of Kaiserslautern
Germany

fieker@mathematik.uni-kl.de

Michael J. Jacobson Jr
University of Calgary
Canada

jacobs@ucalgary.ca

https://doi.org/10.1112/S1461157016000358 Published online by Cambridge University Press