Deuring for the People: Supersingular Elliptic Curves with Prescribed Endomorphism Ring in General Characteristic

Jonathan Komada Eriksen[1], Lorenz Panny[2], Jana Sotáková[3], and Mattia Veroni[1]

[1] Norwegian University of Science and Technology
jonathan.k.eriksen@ntnu.no, mattia.veroni@ntnu.no
[2] Academia Sinica, Taipei, Taiwan
lorenz@yx7.cc
[3] University of Amsterdam and QuSoft, Amsterdam, The Netherlands
ja.sotakova@gmail.com

Abstract. Constructing a supersingular elliptic curve whose endomorphism ring is isomorphic to a given quaternion maximal order (one direction of the *Deuring correspondence*) is known to be polynomial-time assuming the generalized Riemann hypothesis [KLPT14; Wes21], but notoriously daunting in practice when not working over carefully selected base fields.
In this work, we speed up the computation of the Deuring correspondence in *general* characteristic, i.e., without assuming any special form of the characteristic. Our algorithm follows the same overall strategy as earlier works, but we add simple (yet effective) optimizations to multiple subroutines to significantly improve the practical performance of the method.
To demonstrate the impact of our improvements, we show that our implementation achieves highly practical running times even for examples of cryptographic size. One implication of these findings is that cryptographic security reductions based on KLPT-derived algorithms (such as [EHLMP18; Wes22]) have become tighter, and therefore more meaningful in practice.
Another is the pure bliss of fast(er) computer algebra: We provide a Sage implementation which works for general primes and includes many necessary tools for computational number theorists' and cryptographers' needs when working with endomorphism rings of supersingular elliptic curves. This includes the KLPT algorithm, translation of ideals to isogenies, and finding supersingular elliptic curves with known endomorphism ring for general primes.
Finally, the Deuring correspondence has recently received increased interest because of its role in the SQISign signature scheme [DeF+20]. We provide a short and self-contained summary of the state-of-the-art algorithms without going into any of the cryptographic intricacies of SQISign.

Keywords: Algorithms, supersingular elliptic curves, endomorphism rings, quaternion algebras

1 Introduction

Every supersingular elliptic curve defined over a field of characteristic p has endomorphism ring isomorphic to a maximal order in a quaternion algebra ramified only at p and ∞. Conversely, for every maximal order in such a quaternion algebra, there exists a supersingular elliptic curve whose endomorphism ring is isomorphic to this order. This correspondence is called the Deuring correspondence (see Section 2.5 for a precise formulation) and is an important tool in isogeny-based cryptography.

The Deuring correspondence allows us to translate problems which are assumed to be hard for elliptic curves into analogous questions about maximal orders in quaternion algebras, which are often more tractable. For instance, while finding smooth degree isogenies between supersingular elliptic curves over $\mathbb{F}_{p^2}$ is assumed to be hard, the analogous problem for quaternionic orders can be solved in polynomial time with the KLPT algorithm [KLPT14]. The security of virtually all isogeny-based cryptography relies on the hardness of computing the endomorphism ring of a supersingular elliptic curve (the Deuring correspondence in one direction).

The other direction — constructing a supersingular elliptic curve with a given endomorphism ring — is called *Constructive Deuring Correspondence*. It is known to be computable in polynomial time assuming

*Author list in alphabetical order; see https://ams.org/profession/leaders/CultureStatement04.pdf.
This research was funded in part by the Dutch Research Council (NWO) through Gravitation-grant Quantum Software Consortium – 024.003.037. Date of this document: 2023-02-02.

the generalized Riemann hypothesis [KLPT14; Wes21]. Recently, the Constructive Deuring Correspondence has been used constructively in the post-quantum isogeny-based cryptographic signature scheme SQISign [DeF+20; DLW22]. However, the signature scheme SQISign is only implemented for certain primes p of a very special form. In this paper, we revisit the problem of computing the Constructive Deuring Correspondence for all primes p.

Previous work. Early algorithms to find a supersingular elliptic curve with a specified endomorphism ring required exponential time [Cer04; CG14]. With the introduction of the KLPT algorithm [KLPT14], it became possible to solve this problem in heuristic polynomial time, as described in [EHLMP18]: the KLPT algorithm produces an *ideal* connecting the given order to the endomorphism ring of some well-chosen elliptic curve E_0, and this ideal is then translated to an isogeny whose codomain E is the desired curve. Wesolowski [Wes21] later gave a variant of the KLPT algorithm which is provably polynomial-time assuming GRH, resting the algorithm on more solid theoretical foundations and leading to more security reductions between related problems in isogeny-based cryptography [Wes22].

Despite these groundbreaking implications, earlier efforts to implement the KLPT algorithm had suggested that computations relying on KLPT could be largely impractical for parameter sizes relevant for isogeny-based cryptography: the main bottleneck is the *ideal-to-isogeny* translation, that is, translating the KLPT output (a quaternionic ideal of smooth norm) to a sequence of computable isogenies. The exception is in the case when the characteristic p is chosen to be especially nice (that is, such that p^2-1 has a large smooth factor), such as in SQISign [DeF+20; DLW22]. The case of general characteristic (without any conditions on the prime p) was studied in at least two earlier works [Ray18; Kam+22]. In [Ray18], the focus was on expository aspects rather than fast implementation, and even examples with p as small as 1619 required several minutes for the ideal-to-isogeny translation. The approach of [Kam+22] is practical for larger sizes, but their ideal-to-isogeny step involves precomputing certain symbolic formulae for isogenies, which are currently only available up to degree 131 and grow very quickly in general, so [Kam+22] only covers primes up to about 25 bits. We note that these implementations all restrict to the case $p \equiv 3 \pmod 4$.

Contributions. In this work, we devise an algorithm to compute the Deuring correspondence in general characteristic — that is, without assuming any special form of the prime p. One of the simplest and most effective optimizations in our implementation comes from the observation that there is a trade-off between the degrees of the isogenies we use, and extension fields needed to compute such isogenies. We optimize for keeping the degree of the extensions low, allowing for isogenies of larger prime-power degree. In practice, we take a *cost model* as input (describing the contribution of each prime power), and use a greedy algorithm to find the best configuration of degrees which minimizes cost while keeping the total degree large enough for the KLPT algorithm. This simple improvement makes the algorithm much faster in practice: Our implementation (in SageMath) computes the Deuring correspondence for generic 200-bit primes in less than an hour on a single CPU core.

Building upon results of Ibukiyama [Ibu82] and Bröker [Brö09], we present an algorithm to construct supersingular elliptic curves over $\mathbb{F}_p$ together with explicitly known (*effective*) endomorphism rings, for any p. The outline of our algorithm was previously known, but we optimize one crucial subroutine, which results in striking practical speedups when compared to earlier methods.

We speed up the computation of the ideal-to-isogeny translation step with the help of two improved algorithms: We compute the ideal kernel by a new method that completely avoids point divisions and discrete-logarithm computations, and we give a faster algorithm for computing the kernel polynomial of a rational isogeny when given a generating irrational point.

Additionally, our method "automatically" exploits the particular structure of the primes typically used in isogeny-based cryptography. Cryptographic protocols like SQISign typically work with primes p such that p^2-1 contains a large smooth factor, so that all individual isogeny steps can be computed over $\mathbb{F}_{p^2}$. Our method extends this approach, and even though our general implementation cannot compete with the optimized SQISign implementation, it is able to practically compute with a 256-bit prime that has been suggested for use in SQISign.

Our implementation works for arbitrary p without any congruence conditions. To the best of our knowledge, this is the first implementation for primes $p \not\equiv 3 \pmod 4$.

Code. https://github.com/friends-of-quaternions/deuring

1.1 Organization of the paper

The paper is organised as follows:

- In Section 2 we recall some notions on supersingular elliptic curves, isogenies and quaternion algebras, concluding the section with the constructive Deuring correspondence;
- In Section 3 we recall the steps of the standard approach to computing the Deuring correspondence;
- In Section 4 we discuss our improvements, applying optimizations known from other contexts as well as introducing new algorithmic techniques to generalize and accelerate the computation;
- In Section 5 we present empirical timings for our implementation, clearly demonstrating its applicability to cryptographically-sized parameters, and discuss some numerical examples.

2 Preliminaries

In this section we recall some basic notions on supersingular elliptic curves, isogenies and quaternion algebras, concluding with the Deuring correspondence. We refer the interested reader to [Sil09] and [Voi21] for detailed accounts of elliptic curves and quaternion algebras respectively.

Throughout, the letter p will denote a prime integer greater than 3.

Denote by $f^{O(1)}$ the set of functions bounded above by some polynomial in f. The "soft-O" notation $\widetilde{O}(f)$ is shorthand for $f \cdot (\log f)^{O(1)}$.

An integer N is *B-smooth* if none of its prime factors are larger than B. For brevity, we say that N is *smooth* if it is B-smooth for some $B \in (\log p)^{O(1)}$.

We let $M(k)$ denote the cost of arithmetic on polynomials over $\mathbb{F}_{p^2}$ of degree bounded by k; computing operations in $\mathbb{F}_{p^{2k}}$ has the same cost. The standard asymptotics are quadratic time $M(k) \in O(k^2) \cdot M(1)$ for naïve "schoolbook" arithmetic and quasilinear time $M(k) \in O(k \log k \log\log k) \cdot M(1)$ for FFT-based "fast" arithmetic [CK91].

2.1 Isogenies of elliptic curves over finite fields

Every elliptic curve E over $\mathbb{F}_q$ of characteristic $p > 3$ admits a short Weierstraß equation $y^2 = x^3 + Ax + B$ with $A, B \in \mathbb{F}_q$. The set of $\mathbb{F}_q$-rational points of E is defined as

$$E(\mathbb{F}_q) = \{(x, y) \in (\mathbb{F}_q)^2 : y^2 = x^3 + Ax + B\} \cup \{0_E\}$$

where 0_E is the *point at infinity*. This set is a finite abelian group with respect to elliptic-curve point addition and 0_E the neutral element. The *discriminant* of E is the quantity $\Delta(E) := -16(4A^3 + 27B^2)$, and the *$j$-invariant* of E is $j(E) := -1728/\Delta(E)$. Two curves are isomorphic over $\overline{\mathbb{F}}_p$ if and only if their j-invariants are equal. A *twist over* $\mathbb{F}_q$ of $E/\mathbb{F}_q$ is another elliptic curve $\widetilde{E}/\mathbb{F}_q$ with $j(\widetilde{E}) = j(E)$, which is not isomorphic to E over $\mathbb{F}_q$. The curve $\widetilde{E}$ is a *quadratic twist* of E if $\widetilde{E}$ is a twist of E that is isomorphic to E over a quadratic extension of $\mathbb{F}_q$ (but not over $\mathbb{F}_q$).

Given two elliptic curves E and E' over $\mathbb{F}_q$, an *isogeny* $\varphi\colon E \longrightarrow E'$ over $\mathbb{F}_q$ is a morphism over $\mathbb{F}_q$ mapping the identity of E into the identity of E'. Two curves E, E' are *isogenous* over $\mathbb{F}_q$ if there exists an isogeny $\varphi\colon E \longrightarrow E'$ over $\mathbb{F}_q$. By Tate's theorem, we know that E, E' are isogenous over $\mathbb{F}_q$ if and only if they have the same number of $\mathbb{F}_q$-rational points.

Any finite subgroup K of E defines an isogeny $\varphi\colon E \longrightarrow E'$ whose kernel equals K. The isogeny φ can be defined over the same field as the subgroup K and is unique up to post-composition with purely inseparable isogenies (in particular, isomorphisms). The *degree* of such an isogeny is its degree as a morphism, and is equal to the size of its kernel. Degrees are multiplicative with respect to isogeny composition. Given an isogeny $\varphi\colon E \longrightarrow E'$ of degree d over $\mathbb{F}_{p^k}$, its *dual* $\widehat{\varphi} : E' \longrightarrow E$ is an isogeny of degree d over $\mathbb{F}_{p^k}$ such that $\varphi \circ \widehat{\varphi} = [d]$ on E and $\widehat{\varphi} \circ \varphi = [d]$ on E'.

Note that the even if the isogeny is defined over K, the points in the kernel are not necessarily all K-rational. Therefore, even when working with K-rational isogenies, we will need to be careful about the fields of definition; see Section 2.3. Fortunately, at least for supersingular elliptic curves, the situation is a bit simpler thanks to the power of Frobenius (Section 2.2).

2.2 Frobenius and supersingular curves

An isogeny from E to itself is an *endomorphism*. As usual, we promote the zero map to an endomorphism, so that the set of all endomorphisms (over the algebraic closure $\overline{\mathbb{F}}_p$) of an elliptic curve E form a ring under pointwise addition and composition, called the *endomorphism ring of* E and denoted by $\mathrm{End}(E)$. The ring $\mathrm{End}(E)$ is not always commutative: Over finite fields, the endomorphism ring $\mathrm{End}(E)$ is isomorphic either to an order in an imaginary quadratic extension of $\mathbb{Q}$ or to a maximal order in a quaternion algebra over $\mathbb{Q}$. In the first case E is called *ordinary*, in the latter E is called *supersingular*. Every supersingular elliptic curve in characteristic p is isomorphic to an elliptic curve defined over $\mathbb{F}_{p^2}$.

Any elliptic curve $E/\mathbb{F}_q$ has the *(q-power) Frobenius endomorphism* $\pi\colon E \to E$, $(x, y) \mapsto (x^q, y^q)$. Part of its significance lies in the fact that the number of $\mathbb{F}_q$-rational points on E is given by the formula $\#E(\mathbb{F}_q) = q+1-\mathrm{tr}(\pi)$, where the *trace* $\mathrm{tr}(\vartheta)$ of any endomorphism ϑ is defined as the quantity $\vartheta+\widehat{\vartheta} \in \mathbb{Z}$. Quadratic twisting negates Frobenius.

Group structure. Going even further, the structure of the group of rational points on supersingular elliptic curves can be characterized almost exactly. We make use of the following properties:

Lemma 1. *Let E be a supersingular elliptic curve defined over $\mathbb{F}_p$, for $p > 3$ prime. Then the p-power Frobenius π of E satisfies $\pi^2 = -p$ and*

$$|E(\mathbb{F}_{p^{2k}})| = \left(p^k - (-1)^k\right)^2.$$

Proof. To count the number of points over $\mathbb{F}_{p^{2k}}$, we consider the $2k$-th power of the p-power Frobenius π. From supersingularity and Hasse's bounds we have $\pi^2 = -p$, so $\pi^{2k} = (-p)^k$, which has trace $2(-p)^k$. Therefore, $\#E(\mathbb{F}_{p^{2k}}) = p^{2k} + 1 - 2(-p)^k = ((-p)^k - 1)^2 = (p^k - (-1)^k)^2$. □

Theorem 2. *Let E be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$ such that the p^2-power Frobenius π equals $-p$. Then*

$$E(\mathbb{F}_{p^{2k}}) \;\cong\; \mathbb{Z}/(p^k - (-1)^k) \oplus \mathbb{Z}/(p^k - (-1)^k)\,.$$

Moreover, the quadratic twist over $\mathbb{F}_{p^{2k}}$ of E satisfies

$$\widetilde{E}(\mathbb{F}_{p^{2k}}) \;\cong\; \mathbb{Z}/(p^k + (-1)^k) \oplus \mathbb{Z}/(p^k + (-1)^k)\,.$$

Proof. See [Sch87, Lemma 4.8(ii)] or [Len96, Theorem 1(b)]. □

As a consequence, for such a curve E and arbitrary $N \in \mathbb{Z}_{>0}$, if *any* order-N point on E is rational, then the *entire* N-torsion subgroup (a free $\mathbb{Z}/N$-module of rank 2) is rational over the same field, and the same holds true for the quadratic twist.

Remark 3. In Theorem 2, the order of extending the base field and taking the quadratic twist matters: Twisting over $\mathbb{F}_{p^2}$ and then extending to $\mathbb{F}_{p^{2k}}$ leads to group orders of the form $(p^k - (\pm 1)^k)^2$, while extending first and then twisting over $\mathbb{F}_{p^{2k}}$ leads to (strictly more) orders of the form $(p^k \pm (-1)^k)^2$.

Furthermore, the requirement in Theorem 2 that $\pi = -p$ can (say, in algorithms) always be satisfied by taking an isomorphism:

Lemma 4. *Any supersingular elliptic curve in characteristic $p > 3$ is isomorphic to a curve defined over $\mathbb{F}_{p^2}$ whose Frobenius equals $-p$.*

Proof. As the j-invariant lies in $\mathbb{F}_{p^2}$, we may suppose given a curve $E/\mathbb{F}_{p^2}$. By supersingularity and [Sch87, Theorem 4.2(iii)] the p^2-power Frobenius π of E satisfies $\pi^2 - mp\pi + p^2$ where $m \in \{0, \pm 1, \pm 2\}$; in other words, $\pi = \zeta p$ where ζ is an automorphism of E whose order divides 4 or 6.

On a short Weierstraß equation for E, there exists an element $\alpha \in \overline{\mathbb{F}_p}$ such that $\zeta\colon (x, y) \mapsto (\alpha^2 x, \alpha^3 y)$. Fix any (p^2-1)-th root τ of $-1/\alpha$ and define the (twisting) isomorphism $\psi\colon E \to \widetilde{E}$, $(x, y) \mapsto (\tau^2 x, \tau^3 y)$. An explicit calculation shows that the p^2-power Frobenius on $\widetilde{E}$ is $\widetilde{\pi} = -\psi\zeta^{-1}\pi\psi^{-1} = -p$ as desired. □

One particular consequence of Lemma 4 is that any isogeny between two supersingular elliptic curves can be defined over $\mathbb{F}_{p^2}$, by working with isomorphism representatives on which Frobenius is a scalar. This makes isogenies of supersingular elliptic curves particularly nice to compute with; see Section 2.3.

2.3 Algorithms for computing isogenies

Recall from Section 2.1 that an isogeny is determined, essentially uniquely, by its kernel subgroup. In this section, we will survey methods to compute an isogeny when given a representation of its kernel.

By *computing an isogeny* we refer to a procedure which takes as input an elliptic curve over a finite field $\mathbb{F}_q$ and a representation of the kernel (the specifics vary with the method), and outputs the codomain elliptic curve of an isogeny with the given kernel, along with an efficient algorithm to evaluate the isogeny at points in extensions of $\mathbb{F}_q$.

The typical strategy for computing isogenies involves decomposing the isogeny into prime-degree steps for efficiency; in light of this, we shall restrict the discussion below to isogenies $\varphi\colon E \to E'$ of prime degree ℓ. In particular, this entails that the kernel subgroup $K \leq E$ is generated by a single order-ℓ point $P \in E$.

Vélu's formulas. Vélu [Vél71] gave explicit formulas for functions on the domain which are invariant under translations by precisely the kernel subgroup, and which vanish on the kernel points with the correct multiplicities. It is not overly difficult to see (but perhaps somewhat mind-boggling) that such functions are essentially coordinate maps on the quotient E/K, i.e., the isogeny codomain. Interpolating a curve equation is not difficult by evaluating the isogeny at a few points, but there are even general formulas in terms of the x-coordinates of kernel points.

Computationally, the result is an algorithm for evaluating the isogeny at a point that involves iterating over points in the kernel subgroup and performing elliptic-curve group operations between the kernel points and the evaluation point. In particular, the computations must be performed in a ring containing both the kernel points *and* the evaluation point: In the typical case of finite fields the degree of this compositum equals the least common multiple of the individual degrees. (Recall that the field of definition of the *points* inside K may be much larger than the field of definition of K, or equivalently φ.)

The time required to evaluate Vélu's formulas is $O(\ell)$ operations in the base field. This complexity was subsequently improved by (essentially) a square-root factor: The $\sqrt{\text{élu}}$ algorithm from [BDLS20] achieves the same result using only $\widetilde{O}(\sqrt{\ell})$ operations by exploiting a baby-step-giant-step decomposition of the kernel subgroup and quasilinear-time "elliptic resultant" computations. In practice, $\sqrt{\text{élu}}$ begins to outperform Vélu's formulas starting from $\ell \approx 100$.

Our application requires computing isogenies whose kernel points (despite defining an $\mathbb{F}_{p^2}$-rational isogeny) lie in various extension fields $\mathbb{F}_{p^{2k}}$ of $\mathbb{F}_{p^2}$, and evaluating them at points which may lie in *different* extension fields $\mathbb{F}_{p^{2k'}}$ of $\mathbb{F}_{p^2}$. Vélu's formulas must thus be applied over $\mathbb{F}_{p^{2k}} \otimes \mathbb{F}_{p^{2k'}} = \mathbb{F}_{p^{2\,\mathrm{lcm}(k,k')}}$. The cost of working in these field extensions depends on the extension degrees k, k' in a crucial way.

We shall see below that this issue can be (partially) remedied by using a different approach to isogeny evaluation: Instead of working with individual kernel points, one starts from (the radical of) the denominator of (the rational form of) the isogeny, which encodes information about all kernel points at once — and, very conveniently, has coefficients in the field of definition of the isogeny. In our application this permits reducing the required field extensions from degree $\mathrm{lcm}(k, k')$ to degrees k, k' separately.

Kernel polynomials. Every $\mathbb{F}_q$-rational isogeny $\varphi\colon E \to E'$ between two short Weierstraß curves has a standard representation given by rational functions

$$(x, y) \longmapsto \big(f(x),\, cyf'(x)\big)$$

where $f \in \mathbb{F}_q(X)$ is a rational function, f' its formal derivative, and $c \in \mathbb{F}_q^\times$ a nonzero constant; see [Gal12, Theorem 9.7.5]. Writing $f = f_1/f_2$ with coprime polynomials $f_1, f_2 \in \mathbb{F}_q[X]$, the nonzero points P lying in the kernel of φ are characterized by $f_2(x(P)) = 0$. The *kernel polynomial* h of φ is the radical of f_2.

More concretely, the *kernel polynomial* defining a finite subgroup $K \leq E$, or an isogeny with kernel K, is the unique monic squarefree polynomial h_K whose set of roots is precisely the set of x-coordinates of nonzero points in K. Partitioning K as $K = S_2 \sqcup S \sqcup (-S) \sqcup \{0_E\}$ where $S_2 \subseteq K$ is the (possibly empty) subset of points in K of order 2, the kernel polynomial equals $h_K = \prod_{P \in S \cup S_2} (X - x(P))$. Note that whenever K is a subgroup defined over $\mathbb{F}_q$, then (as a result of K being closed under the action of the Galois group) the coefficients of h_K are in $\mathbb{F}_q$ as well, even if its roots $x(P)$ lie outside of $\mathbb{F}_q$.

To compute the kernel polynomial when K is given as a single generating point P, the simplest approach is to iteratively enumerate the points $P, [2]P, \ldots, [\lfloor \ell/2 \rfloor]P$ using repeated point additions and

then compute the kernel polynomial $h_K = \prod_{i=1}^{\lfloor \ell/2 \rfloor} (X - x([i]P))$ with a product tree; this takes time $\widetilde{O}(\ell)$ in the field of definition of P. For a slightly more efficient method, see Algorithm 4.

Kohel's formulas. Kohel [Koh96, § 2.4] gave an algorithm to compute an isogeny from its domain curve and kernel polynomial. The main idea is the following: the rational functions defining φ must satisfy a short Weierstraß equation — that of the codomain. Writing $f = f_1/f_2$ for the x-coordinate map of the isogeny as above and setting $c = 1$, we thus get a differential equation $(X^3 + AX + B)f'(X)^2 = f(X)^3 + \widetilde{A}f(X) + \widetilde{B}$ with unknowns $\widetilde{A}, \widetilde{B} \in \mathbb{F}_q$ and $f \in \mathbb{F}_q(X)$, and where $E\colon y^2 = x^3 + Ax + B$. Moreover, by assumption, the denominator f_2 of f must have the same roots as the kernel polynomial h.

Kohel's formulas then give a solution to this problem: one obtains simple algebraic formulas for $\widetilde{A}, \widetilde{B}$ in terms of the coefficients of E and h, and algebraic formulas for f_1 and f_2 in terms of the coefficients of E, the kernel polynomial h, and its derivatives h', h''. All polynomials appearing in the formulas have degree $O(\ell)$, which implies that they can be evaluated within $\widetilde{O}(\ell)$ base-field operations using FFT-based polynomial arithmetic. For a more elaborate discussion of the complexity, see [Shu09, Theorem 3.1.10].

Irrational $\sqrt{élu}$. Kohel's formulas work with the kernel polynomial, which has size $O(\ell)$: its appearance immediately thwarts all hope for achieving complexity sublinear in ℓ. However, for prime ℓ, the kernel is uniquely defined by any irreducible divisor of the kernel polynomial; we exploit this in Algorithm 3. An algorithm which interpolates between the $\sqrt{\text{élu}}$ and Kohel approaches by working with irreducible (rational!) divisors of the kernel polynomial is described in [BDLS20, § 4.14], where it is argued that this algorithm cannot be expected to improve upon Kohel's for average inputs. However, the approach seems well-suited for the particular situation where the irreducible divisors have degree significantly smaller than $\sqrt{\ell}$, assuming a suitable index system can be found. As far as we know, this variant of the algorithm has never been implemented.

Notation for isogenies. By abuse of language, one often refers to "the" isogeny defined by a finite subgroup K, and "the" target curve is often denoted by E/K, to emphasize that the kernel is K. However, the curve E/K is only defined up to post-composing with an isomorphism, and so this notation mixes models of curves with isomorphism classes. However, this notation is wide-spread, and for computational purposes, we will always understand E/K as being computed from K using Vélu's or Kohel's formulas.

2.4 Quaternion algebras

A *quaternion algebra* B over $\mathbb{Q}$ is a four-dimensional central simple algebra over $\mathbb{Q}$. Every quaternion algebra admits a $\mathbb{Q}$-basis $\{1, \mathbf{i}, \mathbf{j}, \mathbf{k}\}$ with $\mathbf{ij} = -\mathbf{ji} = \mathbf{k}$ and $\mathbf{i}^2 = -q$, $\mathbf{j}^2 = -p$ where $q, p \in \mathbb{Q}^\times$; we write $B = (-q, -p \mid \mathbb{Q})$. Every quaternion $\alpha = t + x\mathbf{i} + y\mathbf{j} + z\mathbf{k} \in B$ has a *conjugate* $\overline{\alpha} := t - x\mathbf{i} - y\mathbf{j} - z\mathbf{k} \in B$; conjugation is an involution. From this, one can define the *reduced trace* and the *reduced norm* as:

$$\begin{aligned} \mathrm{trd}(\alpha) &:= \alpha + \overline{\alpha} = 2t \\ \mathrm{nrd}(\alpha) &:= \alpha\overline{\alpha} = t^2 + qx^2 + py^2 + pqz^2. \end{aligned}$$

Now consider a prime ℓ. The quaternion algebra $B_\ell := B \otimes_{\mathbb{Q}} \mathbb{Q}_\ell$ is obtained by extending the scalars of B from $\mathbb{Q}$ to $\mathbb{Q}_\ell$. This notation includes ∞ by setting $B_\infty := B \otimes_{\mathbb{Q}} \mathbb{R}$. We say that B is ramified at ℓ (including $\ell = \infty$) if B_ℓ is a division ring. We will only consider the quaternion algebra $B_{p,\infty}$ ramified at p and ∞, since the endomorphism ring of a supersingular elliptic curve over a field of characteristic p is isomorphic to an order in this quaternion algebra.

Orders and ideals. A *fractional ideal* I of B is a $\mathbb{Z}$-lattice contained in B, which can be written as $I = \alpha_1\mathbb{Z} + \alpha_2\mathbb{Z} + \alpha_3\mathbb{Z} + \alpha_4\mathbb{Z}$ for a $\mathbb{Q}$-basis $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ of B. The norm of a fractional ideal is defined as $\mathrm{nrd}(I) = \gcd(\{\mathrm{nrd}(\alpha) : \alpha \in I\})$; note that it suffices to evaluate the gcd on a generating set. An *order* is a fractional ideal that is also a subring of B. An order $\mathcal{O}$ is *maximal* if, for any other order $\mathcal{O}'$, we have that $\mathcal{O} \subseteq \mathcal{O}'$ implies $\mathcal{O} = \mathcal{O}'$. For every fractional ideal I in B one can define the *left order* $\mathcal{O}_L(I) := \{\beta \in B : \beta I \subseteq I\}$ and the *right order* $\mathcal{O}_R(I) := \{\beta \in B : I\beta \subseteq I\}$. Saying that I is a *left $\mathcal{O}$-ideal* means that $\mathcal{O} \subseteq \mathcal{O}_L(I)$, and saying it is a *right $\mathcal{O}'$-ideal* means that $\mathcal{O}' \subseteq \mathcal{O}_R(I)$, in which case it is clear that I is a $\mathcal{O}$-$\mathcal{O}'$-bimodule. Two left $\mathcal{O}$-ideals I, J in $B_{p,\infty}$ are *(right) equivalent* if $J = I\beta$ for some $\beta \in B_{p,\infty}^\times$. In this case, $\mathcal{O}_R(I) \cong \mathcal{O}_R(J)$, with conjugation by β defining an isomorphism.

A fractional ideal is *integral* if it is contained in its left (or equivalently right) order. If a left fractional $\mathcal{O}$-ideal I is integral, it is also a left $\mathcal{O}$-ideal in the usual sense, hence we often simply refer to integral ideals as ideals. These ideals have integer norm and can be written as $I = \mathcal{O}_L(I)\alpha + \mathcal{O}_L(I)\mathrm{nrd}(I)$ for any $\alpha \in \mathcal{O}_L(I)$ satisfying $\gcd(\mathrm{nrd}(\alpha), \mathrm{nrd}(I)^2) = \mathrm{nrd}(I)$, or analogously with their right orders. We say that two orders $\mathcal{O}$ and $\mathcal{O}'$ are *connected* if there exists an invertible ideal I with $\mathcal{O}_L(I) = \mathcal{O}$ and $\mathcal{O}_R(I) = \mathcal{O}'$, and we call I a *connecting* $(\mathcal{O}, \mathcal{O}')$-ideal.

Working in a noncommutative ring, the product of ideals is not always well-behaved. Given two ideals $I, J \subseteq B_{p,\infty}$, we say that I is *compatible* with J if $\mathcal{O}_R(I) = \mathcal{O}_L(J)$. If I is compatible with J, then the product $IJ := \{\alpha\beta : \alpha \in I, \beta \in J\}$ is an ideal such that $\mathcal{O}_L(IJ) = \mathcal{O}_L(I)$ and $\mathcal{O}_R(IJ) = \mathcal{O}_R(J)$. Whenever I and J are compatible, the product satisfies $\mathrm{nrd}(IJ) = \mathrm{nrd}(I)\mathrm{nrd}(J)$.

Finally, following [KLPT14, Section 2.3], an order $\mathcal{O} \in B_{p,\infty}$ is *special p-extremal* if it contains a subring $\mathbb{Z}\langle\omega_1, \omega_2\rangle$ with $\mathrm{nrd}(\omega_1) = q$ and $\mathrm{nrd}(\omega_2) = p$ for q coprime to p, and such that the discriminant of $\mathbb{Z}[\omega_1]$ is minimal among all quadratic orders in $B_{p,\infty}$. Note that we can relax the definition and not enforce the smallest discriminant. However, as always for efficiency, we need q small, cf. Section 3.1.

2.5 The Deuring Correspondence

This subsection describes an equivalence between a category from Section 2.1 and another from Section 2.4. The result can be seen as a modern formulation of the so-called Deuring correspondence [Deu41]. For proofs of all statements in this section, see [Voi21, Chapter 42].

In this section we consider two categories. The first one is the category supersingular elliptic curves over $\overline{\mathbb{F}}_p$ under isogenies, denoted as SS_p. As for any supersingular elliptic curve E, its endomorphism ring $\mathrm{End}(E)$ is isomorphic to a maximal order in the quaternion algebra $B_{p,\infty}$. We will see that the homsets $\mathrm{Hom}(E, E')$ between two elliptic curves also carry extra structure.

For two supersingular elliptic curves E, E', fix isomorphisms $\rho\colon \mathcal{O} \xrightarrow{\sim} \mathrm{End}(E)$ and $\rho'\colon \mathcal{O}' \xrightarrow{\sim} \mathrm{End}(E')$ for $\mathcal{O}, \mathcal{O}'$ maximal orders in $B_{p,\infty}$. It is clear that $\mathrm{Hom}(E, E')$ is an abelian group (with addition of isogenies performed pointwise), and further that the action

$$\begin{aligned} \mathrm{Hom}(E, E') \times \mathcal{O} &\to \mathrm{Hom}(E, E') \\ (\phi, \alpha) &\mapsto \phi \circ \rho(\alpha) \end{aligned}$$

turns $\mathrm{Hom}(E, E')$ into a right $\mathcal{O}$-module. With some extra work, one can show that $\mathrm{Hom}(E, E')$ is not only a right $\mathcal{O}$-module, but in fact isomorphic to a right $\mathcal{O}$-ideal. Completely analogously, $\mathrm{Hom}(E, E')$ is isomorphic to a left $\mathcal{O}'$-ideal.

For the second category, let us fix a maximal order $\mathcal{O}_0 \subseteq B_{p,\infty}$, and consider the category of left fractional $\mathcal{O}_0$-ideals under homomorphisms of $\mathcal{O}_0$-modules. Denote this category by $\mathrm{lfrac}\,\mathcal{O}_0$.

By considering a curve E_0 with $\mathrm{End}(E_0) \cong O_0$, and passing to the homsets, we get the functor

$$\mathrm{Hom}(-, E_0) : \mathrm{SS}_p \to \mathrm{lfrac}\,\mathcal{O}_0.$$

This functor is actually an equivalence of categories [Koh96, Theorem 45], which has many important consequences. For instance, we have noted before that every supersingular curve has endomorphism ring isomorphic to a maximal order in a quaternion algebra $B_{p,\infty}$. But from this equivalence of categories, and since every pair of maximal orders has a connecting ideal, the converse is in fact also true: for every maximal order $\mathcal{O} \subseteq B_{p,\infty}$, there exists a supersingular curve E defined over $\mathbb{F}_{p^2}$, with $\mathrm{End}(E) \cong \mathcal{O}$. Further, this choice is unique up to Galois conjugacy. This bijection between isomorphism classes of maximal orders in $B_{p,\infty}$ and Galois conjugacy classes of supersingular j-invariants over $\overline{\mathbb{F}}_p$ is one of the classical formulations of the Deuring correspondence.

The inverse of this functor also has a simple description. Suppose given an $\mathcal{O}_0$-ideal I, and assume for simplicity that $p \nmid \mathrm{nrd}(I)$. The ideal I defines the *I-torsion subgroup of E_0*, or *kernel of I*, via

$$E_0[I] = \{P \in E_0 \mid \alpha(P) = 0 \text{ for all } \alpha \in I\},$$

and thereby the *isogeny defined by I*

$$\phi_I\colon E_0 \longrightarrow E_0/E_0[I]$$

whose kernel is $E_0[I]$. The curve $E_I := E_0/E_0[I]$ satisfies $\mathrm{Hom}(E_I, E_0) \cong I$ and we refer to it as the curve corresponding to I. Its endomorphism ring is isomorphic to $O_R(I)$. The isogeny ϕ_I is (by definition)

separable and satisfies $\deg(\phi_I) = \mathrm{nrd}(I)$ and $\phi_{\overline{I}} = \widehat{\phi_I}$. Furthermore, we have $\phi_{IJ} = \phi_J \circ \phi_I$ whenever I and J are compatible and isomorphisms between the quaternion orders and endomorphism rings of the elliptic curves are chosen appropriately.

Nomenclature. Despite being a correspondence, it is customary to refer to the two directions separately: Starting from an elliptic curve, finding its endomorphism ring as a maximal order in the quaternion algebra is typically referred to as *computing the endomorphism ring problem* (we make this precise in Section 2.6). Conversely, starting from a maximal order in a quaternion algebra, the task of finding a (supersingular) elliptic curve with that order as endomorphism ring is called the *constructive Deuring correspondence*. This problem is the main focus of this paper and we give a proper definition in Problem 8.

2.6 Computing with endomorphism rings

The problem of computing the endomorphism ring of an elliptic curve depends on the representation of the endomorphism ring we ask for. This section mostly follows the terminology of [EHLMP18; Wes22]. The basic endomorphism ring representation is as an order in a quaternion algebra:

*Problem 5 (*MaxOrder*).* Given a supersingular elliptic curve $E/\mathbb{F}_{p^2}$, find a maximal order $\mathcal{O} \in B_{p,\infty}$ with an explicit quaternion basis $\alpha_1, \ldots, \alpha_4$ such that $\mathrm{End}(E) \cong \mathcal{O}$.

Having abstract quaternions that generate the endomorphism ring is not the same as being able to compute with the endomorphisms on the curve. We say that an endomorphism $\alpha \in \mathrm{End}(E)$ comes in *efficient representation* if its description has size $(\log p)^{O(1)}$ and we are able to evaluate $\alpha(P)$ for any $P \in E$ in polynomial time (in the size of the input, i.e., the bit length of P).

*Problem 6 (*EndRing*).* Given a supersingular elliptic curve $E/\mathbb{F}_{p^2}$, find endomorphisms $\phi_1, \ldots, \phi_4$ of E in an efficient representation that generate $\mathrm{End}(E)$ as a $\mathbb{Z}$-lattice.

Effective endomorphisms. Answers to Problems 5 and 6 separately are not immediately useful for many advanced computational tasks in supersingular elliptic curves. The strongest form of "knowing the endomorphism ring" involves having solutions to both problems, *and an isomorphism between them.*

Concretely, this data can be represented as a 4-tuple of pairs (α_j, ϕ_j), each containing a quaternion $\alpha_j \in \mathcal{O}$ and an endomorphism $\phi_j \in \mathrm{End}(E)$ in efficient representation, such that mapping $\alpha_j \mapsto \phi_j$ defines a ring isomorphism between $\mathcal{O}$ and $\mathrm{End}(E)$. Whenever a supersingular elliptic curve E comes equipped with this knowledge, we say that E has *effective endomorphism ring.*[4]

Evaluating fractional isogenies. In algorithms involving endomorphism rings, it is common that the endomorphisms giving an effective basis of the endomorphism ring $\phi_1, \ldots, \phi_4$ are not represented directly as rational maps, but as $\mathbb{Q}$-linear combinations of some easy-to-compute endomorphisms $\{\omega_i\}$; see for example [Koh96; BCEMP19; EHLMP20].

In the following, we will work with isogenies given as quotients ψ/t, where $\psi\colon E \to E'$ is an efficiently evaluatable isogeny and t an integer ≥ 1. More formally, ψ/t is shorthand notation for a tensor $\psi \otimes 1/t$ inside $\mathrm{Hom}(E, E') \otimes_{\mathbb{Z}} \mathbb{Q}$. Moreover, we shall assume that t really does divide ψ: we assume that there exists some isogeny $\psi'\colon E \to E'$ such that $\psi \otimes 1/t = \psi' \otimes 1$. If ψ is an endomorphism of E, this divisibility implies that ψ/t is also an endomorphism of E.

In principle, it is possible to compute the rational functions defining ψ/t directly [ML04; McM14], but this usually requires exponential space and time and is therefore not a viable option in most cases.

Luckily, for smooth enough (or otherwise favorable) denominators there is a way to evaluate ψ/t without recovering an explicit representation as rational maps. The basic idea is as follows: To evaluate ψ/t at some point P, it suffices to find any point Q with $[t]Q = P$, amounting to an *elliptic-curve point division*, and simply output $\psi(Q)$. (Proof: $(\psi/t)(P) = (\psi/t)([t]Q) = (\psi/t \cdot t)(Q) = \psi(Q)$.)

The main issue with this approach is that the result of point divisions by t generally live in field extensions of degree linear in t. Decomposing the input point as a sum of points of prime-power order

[4] A very similar, slightly more general notion was called "ε-basis" in [Wes22, Definition 4]; here we leave the isomorphism $\varepsilon\colon B_{p,\infty} \xrightarrow{\sim} \mathrm{End}(E) \otimes \mathbb{Q}$ implicit.

is a way to partially alleviate this [EHLMP18, Algorithm 5]: Suppose P has order n and write u for the largest divisor of t coprime to n. If $u = t$, output $[t^{-1} \bmod n]\psi(P)$. Otherwise, let $\ell_1^{e_1}, ..., \ell_r^{e_r}$ denote the prime powers in the factorization of t/u and write $n = n' \cdot \ell_1^{f_1} \cdots \ell_r^{f_r}$ with $\gcd(n', t/u) = 1$; note that each $f_j \geq 1$. Set $m_1 := n' \cdot \ell_1^{f_1}$ and $m_j := \ell_j^{f_j}$ for $2 \leq j \leq r$. Write P as a sum $P_1 + P_2 + \cdots + P_r$ of points where P_j has order m_j; such points P_j can be found by multiplying P by scalars comprising a *CRT basis* for the sequence $(m_1, ..., m_r)$. Then, for each j, compute Q_j such that $[\ell_j^{e_j}]Q_j = P_j$, and finally output

$$(\psi/t)(P) = \sum_{j=1}^{r} \left[\left(t/\ell_j^{e_j}\right)^{-1} \bmod m_j\right]\psi(Q_j)\,.$$

As a result of applying this technique, the required extensions are now (generally) linear in $\ell_j^{e_j}$, rather than (generally) linear in t, at the expense of requiring more evaluations of ψ.

Remark 7. Any isogeny $\varphi\colon E_0 \to E$ relates the endomorphism rings of E_0 and E via the induced ring embedding [Wat69, § 3] defined by $\mathrm{End}(E) \hookrightarrow \mathrm{End}(E_0) \otimes \mathbb{Q}$, $\alpha \mapsto (\widehat{\varphi} \circ \alpha \circ \varphi)/\deg(\varphi)$. This allows us to represent the basis of $\mathcal{O} \cong \mathrm{End}(E)$ as a rational combination of a basis of $\widehat{\varphi}\mathrm{End}(E_0)\varphi$, albeit with potentially very large denominators. As the algorithm above shows, evaluating endomorphisms represented in this (fractional) way can be prohibitively expensive. However, whenever the endomorphisms in $\mathrm{End}(E_0)$ can be evaluated efficiently and the isogeny φ has powersmooth norm, it can be used to evaluate endomorphisms of E on points of E in polynomial time (in the smoothness bound).

3 Computing the Deuring Correspondence

We refer by the *constructive Deuring correspondence* to the following problem:

*Problem 8 (*Deuring*).* The *Constructive Deuring Correspondence* problem is the following: Given a maximal order $\mathcal{O}$ in $B_{p,\infty}$, compute a supersingular elliptic curve $E/\mathbb{F}_{p^2}$ such that $\mathrm{End}(E) \cong \mathcal{O}$.

As noted before in Section 2.5, the converse to the Deuring is the MaxOrder problem (Problem 5).

Following the KLPT algorithm [KLPT14], we will in fact output E together with a powersmooth isogeny $\phi\colon E_0 \to E$ for a very special E_0 with efficiently represented endomorphism ring $\mathcal{O}_0$. We will construct this E_0 in Section 3.1 and consider it fixed (per characteristic) for the remainder of the discussion. Note that from Remark 7, this means that E will also have an efficient representation of the endomorphism ring $\mathrm{End}(E) \cong \mathcal{O}$.

In light of the categorical equivalence described in Section 2.5, a natural strategy to tackle the Deuring problem for a maximal order $\mathcal{O} \in B_{p,\infty}$ goes as follows:

Step 0: Fix some base curve $E_0/\mathbb{F}_p$ with a known, effective endomorphism ring $\mathcal{O}_0$.
Step 1 (KLPT): Construct an ideal I connecting $\mathcal{O}_0$ and $\mathcal{O}$ of suitable norm.
Step 2 (IdealToIsogeny): Compute the isogeny corresponding to I as $\varphi_I\colon E_0 \to E$.

The target E is the desired curve with $\mathrm{End}(E) \cong \mathcal{O}$.

Steps 1 and 2 are fairly disjoint and algorithmically different steps. However, the complexity of Step 1 needs to be considered together with Step 2: translating quaternionic ideals into isogenies of elliptic curves. For certain ideal norms, the translation into isogenies is easier, and, conversely, it may be infeasible — or even impossible — to find ideals of a specific norm.

We start with Step 0 in Section 3.1; this is done once per characteristic p. Next, upon choosing a target norm R for a given ideal I, Step 1 can be solved using KLPT-like algorithms, as we explain in Section 3.2. We postpone the details on how to select R to Section 4. Finally, we give a high level overview of IdealToIsogeny step in Section 3.3.

3.1 Step 0. Constructing the base curve

For $p \equiv 3 \pmod 4$, it is customary to use the elliptic curve $E_0\colon y^2 = x^3 + x$ with endomorphism ring

$$\mathcal{O}_0 = \mathbb{Z} \oplus \mathbb{Z}\,\mathbf{i} \oplus \mathbb{Z}\,\frac{\mathbf{i}+\mathbf{j}}{2} \oplus \mathbb{Z}\,\frac{1+\mathbf{k}}{2}$$

in $(-1,-p \mid \mathbb{Q})$, where $\mathbf{j}$ corresponds to the p-power Frobenius endomorphism π on E_0 as usual and $\mathbf{i}$ corresponds to the order-4 automorphism $\iota\colon (x,y) \mapsto (-x, \sqrt{-1}\cdot y)$ of E_0. The action of linear combinations of $1, \mathbf{i}, \mathbf{j}, \mathbf{k}$ with denominators can be evaluated using Section 2.6.

Similarly, for $p \equiv 2 \pmod 3$, the standard choice is $E_0\colon y^2 = x^3 + 1$ with endomorphism ring

$$\mathcal{O}_0 \;=\; \mathbb{Z} \oplus \mathbb{Z}\,\frac{1+\mathbf{i}}{2} \oplus \mathbb{Z}\,\frac{\mathbf{j}+\mathbf{k}}{2} \oplus \mathbb{Z}\,\frac{\mathbf{i}+\mathbf{k}}{3}$$

in $(-3,-p \mid \mathbb{Q})$, where $(\mathbf{i}-1)/2$ corresponds to the order-3 automorphism $\omega\colon (x,y) \mapsto (\zeta_3 \cdot x, y)$ of E_0. It is also possible to give the maximal order for $p \equiv 5 \pmod 8$, see [KLPT14, § 2.3]. However, this case is subsumed in the following.

For $p \equiv 1 \pmod 4$, a base curve can be constructed using a combination of Bröker's algorithm [Brö09] with a classification result from Ibukiyama [Ibu82, Theorem 1] on quaternion maximal orders containing a norm-p element; see also [KLPT14, § 2.3] and [EHLMP18, § 5.1]. The steps are as follows: Find the smallest prime $q \equiv 3 \pmod 4$ such that p is ramified or inert in $\mathbb{Q}(\sqrt{-q})$, compute the unique [CX22] root $j \in \mathbb{F}_p$ of the *Hilbert class polynomial* H_{-q}, and construct $E_0/\mathbb{F}_p$ with j-invariant j. The endomorphism ring of E_0 is isomorphic to a maximal order in the quaternion algebra $B_{p,\infty} = (-q,-p \mid \mathbb{Q})$, hence $\mathbf{i}$ corresponds to an endomorphism ϑ of E_0 such that $\vartheta^2 = [-q]$ and $\vartheta\pi = -\pi\vartheta$.

As q is tiny, one can find ϑ explicitly by simply enumerating and testing all q-isogenies $E_0 \to E_0$; this method is already polynomial-time in q. However, there is a faster way: Fixing a short Weierstraß model $E_0\colon y^2 = x^3 + ax + b$ with $a, b \in \mathbb{F}_{p^2}$, the curve $E_0'\colon y^2 = x^3 + q^2ax - q^3b$ is the codomain of the isomorphism $\tau\colon E_0 \to E_0'$ given by $(x,y) \mapsto (-qx, \sqrt{-q}^3 y)$. The desired endomorphism $\vartheta\colon E_0 \to E_0$ acts on the standard Weierstraß differential $\mathrm{d}x/y$ via multiplication by $\sqrt{-q} \in \mathbb{F}_{p^2}$, while τ acts as $1/\sqrt{-q}$ by construction. Hence, the composition $\vartheta' = \tau\vartheta\colon E_0 \to E_0'$ is a normalized isogeny of degree q, which can be computed within $\widetilde{O}(q)$ operations in $\mathbb{F}_p$ using [BMSS08]. Then clearly $\vartheta = \tau^{-1}\vartheta'$. Choosing the other square root of $-q$ in the definition of τ recovers $\widehat{\vartheta} = -\vartheta$, which is also a correct output.

Then, according to Ibukiyama, there are only two candidates for $\mathrm{End}(E_0)$, namely

$$\mathcal{O}_0 \;=\; \mathbb{Z} \oplus \mathbb{Z}\,\frac{1+\mathbf{i}}{2} \oplus \mathbb{Z}\,\frac{\mathbf{j}+\mathbf{k}}{2} \oplus \mathbb{Z}\,\frac{c\mathbf{i} \pm \mathbf{k}}{q} \tag{1}$$

where c is a fixed integer satisfying $c^2 \equiv -p \pmod q$. The correct choice of sign can be determined by evaluating the endomorphism $\vartheta([c] + \pi)$ associated to $c\mathbf{i} + \mathbf{k}$ on a basis of the q-torsion of E_0: The $+$ is correct if the image is trivial, $-$ otherwise.

Remark 9. The curve E_0 is the reduction modulo p of an elliptic curve in characteristic zero with *complex multiplication* by the imaginary quadratic ring $\mathbb{Z}\big[(1+\sqrt{-q})/2\big]$. Note that the degree $\deg(H_{-q}) \approx \sqrt{q}$, which can quickly get expensive to compute with. Fortunately, assuming GRH, the minimal q is in $O((\log p)^2)$ and can in practice be found very easily.

Finally, notice that the above demonstrates that constructing supersingular elliptic curves in this way reveals the endomorphism ring, as [CPV20; LB20] showed previously. Therefore, this method is not suited to solve the open problem of *hashing into the supersingular isogeny graph*, see for instance [Boo+22].

Representation of quaternion orders. Choosing $\mathcal{O}_0$ in the way described above means we will work in the quaternion algebra $(-q,-p \mid \mathbb{Q})$. However, the quaternion order $\mathcal{O}$ given to us may have been represented as an order in a different, but isomorphic quaternion algebra $(-q',-p' \mid \mathbb{Q})$: For instance, for $p \equiv 11 \pmod{12}$ we could use either of the constructions for $p \equiv 3 \pmod 4$ or $p \equiv 2 \pmod 3$ to express $\mathcal{O}_0$. Both choices are natural as they correspond to the well-known curves $E_{1728}\colon y^2 = x^3 + x$ and $E_0\colon y^2 = x^3 + 1$. We will return to this specific situation in Example 22.

In the following, we require that $p = p'$, so that $\mathbf{j}^2 = \mathbf{j}'^2 = -p$. (This holds true for all constructions of $\mathcal{O}_0$ given above.) Then we can pass between the two representations of the quaternion algebra $B_{p,\infty}$ using the following lemma:

Lemma 10. *Let p be a prime number and $q, q' \in \mathbb{Z}_{>0}$ such that $B = (-q,-p \mid \mathbb{Q})$ and $B' = (-q',-p \mid \mathbb{Q})$ are quaternion algebras ramified at p and ∞.*

Then there exist $x, y \in \mathbb{Q}$ such that $x^2 + py^2 = q'/q$. Writing $1, \mathbf{i}', \mathbf{j}', \mathbf{k}'$ for the generators of B' and $1, \mathbf{i}, \mathbf{j}, \mathbf{k}$ for the generators of B, and setting $\gamma := x + y\mathbf{j}$, the mapping

$$\mathbf{i}' \mapsto \mathbf{i}\gamma, \qquad \mathbf{j}' \mapsto \mathbf{j}, \qquad \mathbf{k}' \mapsto \mathbf{k}\gamma$$

defines a $\mathbb{Q}$-algebra isomorphism $B' \xrightarrow{\sim} B$.

Proof. Existence of (x, y): Since B and B' are ramified at the same places, there exists an isomorphism $f\colon B' \to B$. By the Skolem–Noether theorem, we may without loss of generality assume $f(\mathbf{j}') = \mathbf{j}$; see for instance [Voi21, Corollary 7.1.5]. Using this, a direct calculation shows $\mathbf{j}f(\mathbf{i}') = -f(\mathbf{i}')\mathbf{j}$, which implies $f(\mathbf{i}') \in \mathbb{Q}\mathbf{i} + \mathbb{Q}\mathbf{k}$. Therefore, the element $\delta := \mathbf{i}^{-1}f(\mathbf{i}')$ is of the form $x + y\mathbf{j}$ with $x, y \in \mathbb{Q}$, and we have $x^2 + py^2 = \mathrm{nrd}(\delta) = q'/q$ by multiplicativity of the norm.

Correctness of the constructed isomorphism is readily verified. □

The sledgehammer method to find the pair (x, y) in Lemma 10 constructively would consist in running Simon's general algorithm for quadratic forms [Sim05], which runs in polynomial time after factoring q and q', but note that the particular special case required here (a *Legendre equation*) is classical.

3.2 Step 1. Finding a connecting ideal

In this section, we will explain the KLPT step from Section 3. It is based on the KLPT algorithm [KLPT14]; the state-of-the-art improvements were made in the context of SQISign [DeF+20; DLW22].

In the previous section, we have fixed E_0 with an efficient endomorphism ring $\mathcal{O}_0$. Finding a connecting ideal between O_0 and $\mathcal{O}$ is straightforward: set $N = [\mathcal{O}_0 : \mathcal{O} \cap \mathcal{O}_0]$ and then define

$$I(\mathcal{O}_0, \mathcal{O}) := N\mathcal{O}_0\mathcal{O}.$$

It is easy to see that it is a connecting ideal, and it is clearly integral. However, since we do not have any control over N, this choice would almost certainly make all the following steps exponential-time.

From the Deuring correspondence, a curve E with $\mathrm{End}(E) \cong \mathcal{O}$ is defined uniquely up to Galois conjugacy; it corresponds to a left ideal class of $I(\mathcal{O}_0, \mathcal{O})$. Therefore, we search for more suitable ideals among the ideals $I(\mathcal{O}_0, \mathcal{O}) \cdot \beta$ for $\beta \in B_{p,\infty}^{\times}$: They give rise to the same codomain curve (or its conjugate). From now on, all ideals we will discuss will be integral left $\mathcal{O}_0$-ideals in this equivalence class.

The following lemma controls the norm (and works for any order, not just $\mathcal{O}_0$ from Section 3.1).

Lemma 11 ([KLPT14, Lemma 5]). *Let I be a left $\mathcal{O}_0$-ideal and $\alpha \in I$ an element of norm N. Then*

$$\chi_I(\alpha) := I\overline{\alpha}/\mathrm{nrd}(I)$$

is an integral ideal of norm $N/\mathrm{nrd}(I)$.

Therefore, to find an equivalent ideal $I \sim J$ of norm R, one only has to find an element $\alpha \in I$ of norm $\mathrm{nrd}(\alpha) = R \cdot \mathrm{nrd}(I)$. Since I is a 4-dimensional $\mathbb{Z}$-lattice, this is task is equivalent to representing the integer R by a certain positive-definite quadratic form.

Prime norm. Finding an equivalent ideal of prime norm is easy; simply iterate over short vectors in the ideal lattice quickly finds an element $\beta \in I$ such that the norm of $\chi_I(\beta)$ is prime. So from now on, let us assume that the ideal I we start with has prime norm.

Remark 12 (Failure in KLPT). For a random ideal, we expect to find equivalent ideals of prime norm $\approx p^{1/2}$. De Feo, Leroux and Wesolowski [DLW22, Section 3.2] observed that this heuristic may fail if there is a representative in the class of I with unexpectedly small composite norm (smaller than $p^{1/2}$). In our case, this simply means that the KLPT output will have bigger norm than expected, hence we may require more torsion to work with. We fix this by only selecting the target norm (see Section 4.2) after having obtained an equivalent prime ideal. In practice, this almost never happens when working with random ideal classes, except when working with very small primes.

Algorithm 1: $\texttt{KLPT}(\mathcal{O}_0, I, R)$

Input: maximal order $\mathcal{O}_0 \subseteq B_{p,\infty}$, connecting $(\mathcal{O}_0, \mathcal{O})$-ideal I of norm N, target norm R.
Output: $J \sim I$ with $\mathrm{nrd}(J) = R$, or failure.

1 Split R as a product $R = r_1 r_2$, where $r_2 \approx r_1^5$.
2 Find any $\gamma \in \mathcal{O}_0$ of norm Nr_1.
3 Find $\mu_0 \in \mathbf{j}\mathbb{Z}[\mathbf{i}]$ such that $\mathcal{O}_0\gamma\mu_0/N\mathcal{O}_0 = I/N\mathcal{O}_0$.
4 Use strong approximation modulo N on μ_0 to find $\mu \in \mathcal{O}_0$ of norm r_2.
5 Set $\beta = \gamma\mu$ of norm NR.
6 **Return** $\chi_I(\beta) = I\overline{\beta}/N$.

KLPT. Let I be an ideal of prime norm N. The original KLPT algorithm [KLPT14] takes as input a prime ℓ and finds an equivalent ideal $J \sim I$ of norm ℓ^e for some e. It has since been extended to more general norms R, In this section, we will give an overview of the modern formulation of the algorithm: The high-level steps are shown in Algorithm 1.

Following [DeF+20], the substeps are typically called as follows: Step 2 is called RepresentInteger, Step 3 is called IdealModConstraint, and Step 4 is called StrongApproximation. Heuristically, the original KLPT algorithm works as long as R exceeds $\approx p^{7/2}$. Petit and Smith [PS18] improved the StrongApproximation step by searching for a small solution using lattice reduction, rather than returning a random solution, which enables them (and us) to find ideals of norm $\approx p^3$.

To make this algorithm work in heuristic polynomial time, KLPT require $\mathcal{O}_0$ to be a special extremal order. This simplifies the situation in two ways: in Step 2, one can find γ by representing Nr_1 by a quadratic form of the shape $f(t, x) + pf(y, z)$ for a binary quadratic form $f(u, v)$, which allows for reduction to 2 variables and using Cornacchia's algorithm [Cor08]. Note that for general $\mathcal{O}$, even if there exists a suitable decomposition of the quadratic form using a binary form f, the class number of the quadratic order corresponding to f might be too large, and the chances of a random integer being represented by $f(u, v)$ are small. Similarly, using a special extremal order in Step 3 means the search for μ reduces to a search in a quadratic suborder, again making the step much easier.

For our purposes, we have only used the generalization from having the target norm be a power of ℓ to instead be $R = r_1 \cdot r_2 \approx p^3$: In Step 2, one looks for elements of norm Nr_1, and in Step 4 replace the power of ℓ with r_2. Clearly not every choice of r_1, r_2 will work: heuristic estimates suggest that $r_1 \approx p^{1/2}$ and $r_2 \approx p^{5/2}$ should suffice, though if the ideal I is of prime norm $\gg p^{1/2}$ (see Remark 12) then r_2 needs to be bigger as well. Typically, one chooses r_1, r_2 smooth, as we will in Section 4.2.

The KLPT algorithm can further be generalized to a larger class of orders, than just special extremal maximal orders; see [DeF+20; Ler22].

3.3 Step 2. Ideal-to-Isogeny translation

The next step is to translate the ideal J of norm $\mathrm{nrd}(J) = N$ to its corresponding isogeny. Following Section 2.5, one can start by computing the J-torsion subgroup $E_0[J]$. The standard approach is to do so by evaluating the action of J on the N-torsion of E_0. However, the complexity of this approach is in general exponential in $\log(N)$: It follows from Theorem 2 that the torsion group $E_0[N]$ is in general only defined over $\mathbb{F}_{p^{2k}}$ where $k \in O(N)$. Furthermore, computing isogenies of degree N from its kernel group is exponential in $\log(N)$ in general. Therefore, for general norms, translating ideals to isogenies is infeasible.

This is why we need the flexibility of KLPT to efficiently find equivalent ideals of prescribed norm. The simplest way is to set KLPT to target a generic powersmooth norm. However, this is far from optimal, and one of our contributions is precisely an improvement on how to choose this target norm.

To translate the ideal to the corresponding isogeny ϕ_J, we first find the kernel by computing the the J-torsion $E_0[J]$. Algorithms for doing this was first presented by Galbraith, Petit and Silva [GPS17]. Our version is based on this, but include a few tricks we present in Section 4.1. For now, suppose that J has smooth norm $\prod_i \ell_i^{e_i}$. The first step in finding the J-torsion subgroup $E_0[J]$ is to generating the bases of the torsion subgroups $E[\ell_i^{e_i}]$.

Generating bases of torsion groups. We need to generate a basis for the torsion groups $E[\ell_i^{e_i}]$ for all $\ell_i^{e_i} \mid \mathrm{nrd}(J)$. Let k be an integer such $E[\ell_i^{e_i}] \subseteq E(\mathbb{F}_{p^{2k}})$. Generating a basis can be done by sampling random points and multiplying them by a suitable cofactor $(p^k \pm (-1)^k)/\ell_i^{e_i}$; cf. Theorem 2. This in general generates points of order dividing $\ell_i^{e_i}$. With probability $(\ell_i^{2e_i} - \ell_i^{2(e_i-1)})/\ell_i^{2e_i} = (\ell_i^2 - 1)/\ell_i^2$ we obtain a point of full order. However, for two points P, Q to generate $E[\ell_i^{e_i}]$, we also need to check linear independence (for instance, using the Weil pairing to check that the points $[\ell^{e-1}]P$ and $[\ell^{e-1}]Q$ are independent). Hence, we can prove that with overwhelming probability, it is enough to repeat the sampling several times. More importantly, in practice, we only need a few tries.

We see that generating the torsion bases is $O(k \log p) \cdot M(k)$, for $M(k)$ the cost of multiplying in $\mathbb{F}_{p^k}$.

Finding kernel generators. Once we have the bases for $E[\ell^e]$, we can compute $E[J]$. Following [GPS17], the idea is to find the action on the torsion subgroup $E[\ell^e]$ of a set of endomorphisms α_i which generate the ideal J. This is done as follows: Every α_i is a $\mathbb{Q}$-linear combination of the basis $1, \mathbf{i}, \mathbf{j}, \mathbf{k}$ of $B_{p,\infty}$, typically with small denominators, and can hence be evaluated on the basis $\langle P, Q\rangle = E[\ell^e]$ by means of the techniques discussed in Section 2.6. By computing two discrete logarithms (easy if ℓ is small), one finds integers a, b, c, d such that $\alpha_i(P) = [a]P + [b]Q$ and $\alpha_i(Q) = [c]P + [d]Q$, and hence recovers the matrix $M_i \in (\mathbb{Z}/\ell^e)^{2\times 2}$ by which α_i acts on $E[\ell^e] \cong \mathbb{Z}/\ell^e \times \mathbb{Z}/\ell^e$. The intersection of the kernels of all these matrices M_i is (by definition) equal to $E[J]$.

In Section 4.1, we shall present an improved variant of this algorithm which avoids both discrete logarithms and potential point divisions.

Evaluating isogenies. For every prime power $\ell_i^{e_i}$ dividing $\mathrm{nrd}(J)$, we need to compute an isogeny of degree $\ell_i^{e_i}$. In the case that the kernel points found in the previous step are defined over $\mathbb{F}_{p^{2k_i}}$, using Vélu's or Kohel's algorithm (Section 2.3), each isogeny computation has complexity bounded by $\widetilde{O}(\ell_i M(k))$. (Note that we will improve upon this with Algorithm 4).

We use Kohel's algorithm because we need to compute a sequence of isogenies, and evaluate these isogenies at several points defined over different extensions $\mathbb{F}_{p^{2k_j}}$. Using Vélu's or the $\sqrt{\text{élu}}$ formulas would require us to work in the compositum of these two fields, which impacts the performance at least quadratically in the extension degrees.

Remark 13 (Known speedups for computing isogenies). When computing several isogenies of different degrees using points defined over the same extension field, there are many possibilites for small speedups, such as using *(optimal) strategies*. Many of these tricks are successfully used to accelerate the computation of sequences of isogenies in other isogeny protocols, [DJP14; CLMPR18].

In the same spirit, the $\sqrt{\text{élu}}$ formulas offer a significant speedup for isogenies of moderate degree. These formulas can be used directly in the final step of the isogeny evaluation, computing the last isogeny without performing any evaluations and thus saving the cost from possibly having to pass to larger composite extension fields. Even better, we may pick one particular extension field and compute all the isogenies whose generators lie in that field at the very end using $\sqrt{\text{élu}}$. Note that this finds the codomain curve faster in some cases, but evaluating the isogeny itself may become slower.

4 Our improvements

This section explains the specific improvements in our implementation, which works for any prime $p > 3$.

4.1 Computing the kernel

In this section, we go through our algorithm for finding $E[J]$, where J is an ideal of norm $\mathrm{nrd}(J) = N$.

Assuming that J is cyclic (otherwise, we can always scale J by a suitable scalar), we may simplify the algorithm from Section 3.3 by writing $J = \mathcal{O}_0\alpha + \mathcal{O}_0 N$ for some $\alpha \in J$ satisfying $\gcd(\mathrm{nrd}(\alpha), N^2) = N$; see [Ler22, Algorithm 19].

In this case, $E_0[J] = E_0[\alpha] \cap E_0[N]$ and we can easily find $E_0[J]$ as $\overline{\alpha}(E_0[N])$. We do the following: Write $N = \prod \ell_i^{e_i}$. We can evaluate $\overline{\alpha}$ on the bases $\langle P_i, Q_i\rangle = E_0[\ell_i^{e_i}]$ and then take whichever image point has full order (and hence generates the kernel of α restricted to $E_0[\ell_i^{e_i}]$). By the structure theorem for finite abelian groups, these images together generate the kernel of $E_0[\alpha] \cap E_0[N]$. Clearly, this technique avoids discrete-logarithm computations.

Remark 14. We do not have to work on each prime power individually; if one instead wishes to work directly with a basis $\langle P, Q\rangle = E_0[N]$, the group $E_0[J]$ is simply $\langle \overline{\alpha}(P), \overline{\alpha}(Q)\rangle$. (Note that in this case we are not guaranteed that either $\overline{\alpha}(P)$ or $\overline{\alpha}(Q)$ has full order.) This shows that it is easy to find $E_0[J]$ whenever J is cyclic, even if N is not smooth, as long as $E_0[N]$ is defined over a small extension field.

Next, we note how to avoid point divisions. Let α have denominator t. Since $\alpha \in \mathcal{O}_0$, we know that $t \mid 2q$ by construction (see Equation (1)). We will always avoid having to do point division by finding the slightly larger $\ell^{e+\nu_\ell(t)}$-torsion. This only changes the algorithm at two primes at most, that is, 2 and q. The full algorithm is given in Algorithm 2.

Algorithm 2: `IdealToKernelGens`(J, E_0)

Input: left $\mathcal{O}_0$-ideal J of norm $N = \prod_{i=1}^{r} \ell_i^{e_i}$, curve E_0 with effective endomorphism ring $\mathrm{End}(E_0) \cong \mathcal{O}_0$.
Output: $\{G_1, ..., G_r\}$, a generating set of $\ker \phi_I$, with $\mathrm{ord}(G_i) = \ell_i^{e_i}$.

Compute $\alpha \in \mathrm{End}(E_0)$ such that $J = \mathcal{O}_0\alpha + \mathcal{O}_0 N$ under the isomorphism $\mathrm{End}(E_0) \cong \mathcal{O}_0$.
Let $(\phi_1, ..., \phi_4)$ be a basis of $\mathrm{End}(E_0) \cong \mathcal{O}_0$ consisting of efficiently evaluatable endomorphisms.
Write $\overline{\alpha}$ as a fraction of the form $(c_1\phi_1 + ... + c_4\phi_4)/t$, where $c_1, c_2, c_3, c_4 \in \mathbb{Z}$ and $t \in \mathbb{Z}_{\geq 1}$.
For $i \in \{1, \ldots, r\}$ **do**
    Set $v_i = \nu_{\ell_i}(t)$ to be the ℓ_i-adic valuation of t.
    Let $c_j^{(i)} \leftarrow c_j(t/\ell^{v_i})^{-1} \bmod \ell_i^{e_i+v_i}$ for $j \in \{1, ..., 4\}$.
    Define $\gamma_i \leftarrow c_1^{(i)}\phi_1 + \cdots + c_4^{(i)}\phi_4$.
    Find $P, Q \in E_0$ such that $\langle P, Q\rangle = E_0[\ell_i^{e_i+v_i}]$.
    Compute $G_i \leftarrow \gamma_i(P)$.
    If $[\ell_i^{e_i-1}]G_i = 0$ **then**
        Compute $G_i \leftarrow \gamma_i(Q)$.
Return $\{G_1, ..., G_r\}$.

In Algorithm 2, after the basis is found for $E[\ell^{e'}]$ with a suitable e', the cost of finding the kernel is dominated by evaluating the Frobenius endomorphism and is in $O(\log p + e' \cdot \log \ell) \cdot M(k)$. In the typical case that $\ell^{e'}$ is minuscule compared to p, the cost can be simplified to $O(\log p) \cdot M(k)$.

4.2 Choosing the norm

We start with a quote from [DLW22]: "The efficiency of SQISign is mostly governed by the ideal-to-isogeny translation, [...]". However, the cost of this is heavily influenced by the choice of R. In this section, we explain our main trick of choosing R such that this cost is reduced.

We make the following changes to the KLPT algorithm: first, we include the known improvement due to [PS18] in the last step. Second, before running the KLPT algorithm, we include a greedy optimization step, in which we compute the optimal R-torsion to work with. The norm T of the output ideal of the KLPT algorithm will depend on this R.

Selecting favorable torsion. We want to select the best combination of prime-power factors $\ell^e \mid R$ such that the cost of the translation to isogeny step (see Section 3.3) is minimized. It is clear that each prime power $\ell^e \mid R$ contributes in many direct and indirect ways: we need to compute the basis of the ℓ^e-torsion, evaluate the action of the endomorphism ring on this torsion, find the kernel generator, compute up to e different ℓ-isogenies, etc. Moreover, remembering that we use many of the standard implementation tricks such as pushing points through isogenies, the specific amount by which any one prime power is contributing to the total cost is difficult to determine. As such, our implementation takes a simple *cost model* as input, and this cost model estimates the cost of computing with ℓ^e-torsion in an extension of degree k. We then use a greedy algorithm to find $R > B$ for a suitable bound B, depending on the ideal-to-isogeny strategy. When simply aiming to translate the KLPT output directly (as our implementation does by default), the bound is (usually, see Remark 12) $B = p^3$. In Appendix A, we show how this bound can be reduced to $B = p^2$, using a method based on SQISign.

The cost model we use in our implementation works as follows, for some constants $c_1, c_2, c_3, c_4 \in \mathbb{R}_{>0}$:

– the cost ratio of $\mathbb{F}_{p^{2k}}$-operations to $\mathbb{F}_{p^2}$-operations in SageMath (which uses PARI internally) was measured empirically for various sizes of p and k and approximated numerically by simple formulas;
– the cost of computing a basis of $E[\ell^k]$ is modelled as $c_1 \cdot k \cdot \log p$ operations in $\mathbb{F}_{p^{2k}}$;
– the cost of computing the kernel generators in Algorithm 2, done by evaluating the action of the dual of the generator α, is modelled as $c_2 \cdot \log p$ operations in $\mathbb{F}_{p^{2k}}$;
– finally, the cost of computing the ℓ^e-isogeny is modelled as $c_3 \cdot e \cdot \ell \cdot (k + c_4(\log \ell)^2)$ operations in $\mathbb{F}_{p^2}$.

In our experiments, we use $(c_1, c_2, c_3, c_4) = (0.31, 1.17, 0.46, 0.01)$, which were estimated empirically.

We stress that this cost model is very rudimentary. Better results can almost certainly be obtained by fine-tuning this torsion-optimization step.

A picture is worth a thousand words. To illustrate why our approach can lead to improvements, we examine the following two figures. Figure 1 gives the heatmap of the degree of the extensions of $\mathbb{F}_{p^2}$ over which the full ℓ^e-torsion is defined: the intensity of the color of the pixels corresponds to the probability of obtaining degree k. We see that the extension degree is most often $(\ell - 1)/2$.

The naïve/powersmooth strategy picks prime powers ℓ^e starting from the left, and hence usually ends up with extension fields which are in the upper part of the picture (as those are the most probable ones).

Our approach corresponds to looking at a similar picture for a particular p: in Figure 2, each data point corresponds to the extension degree over $\mathbb{F}_{p^2}$ over which the ℓ^e-torsion is defined. We choose data points in the lower part of the picture, skipping degrees which require large extension degrees.

Finally, SQISign chooses for its parameters a prime p such that in the corresponding Figure 2, the bottom of the picture is heavily inhabited, and only compute with torsion which does not require large extension degrees.

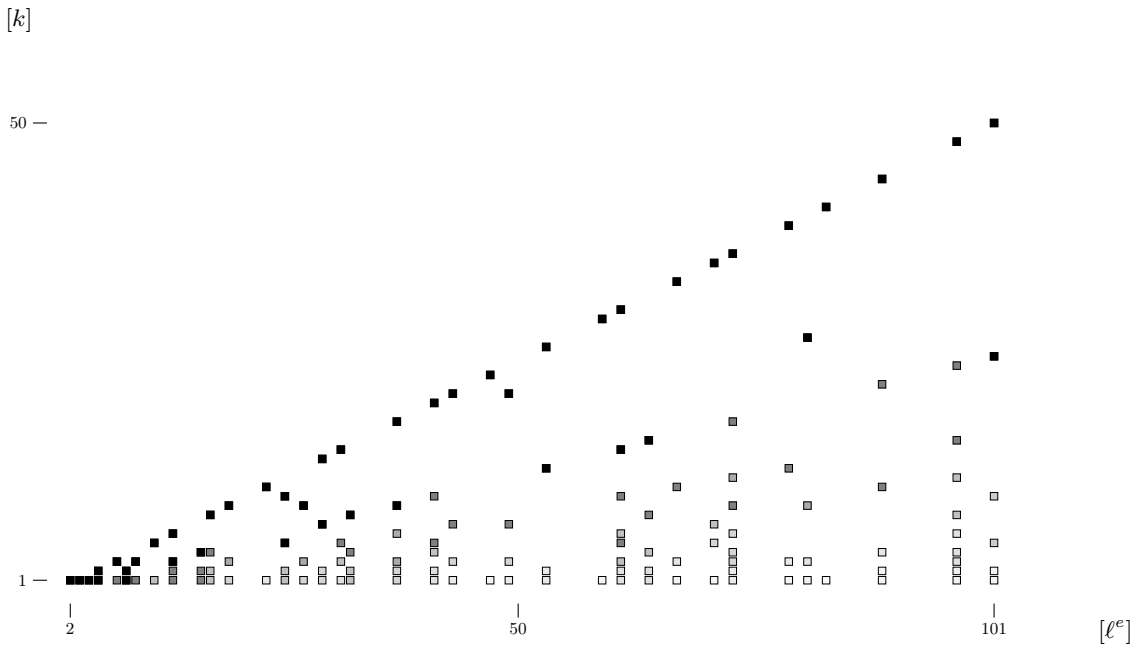


Fig. 1. Heatmap of distribution of extension degrees k required to access the ℓ^e-torsion subgroup of a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$, under the heuristic assumption that p behaves like a random unit modulo all ℓ^e. The intensity with which each data point (ℓ^e, k) is drawn represents the number of units $\mu \in (\mathbb{Z}/\ell^e)^\times$ such that k is minimal with $\mu^k \in \{\pm 1\}$; the choice of sign translates to working on a curve model with Frobenius $(-p)^k$ or $-(-p)^k$ (see Section 4.3).

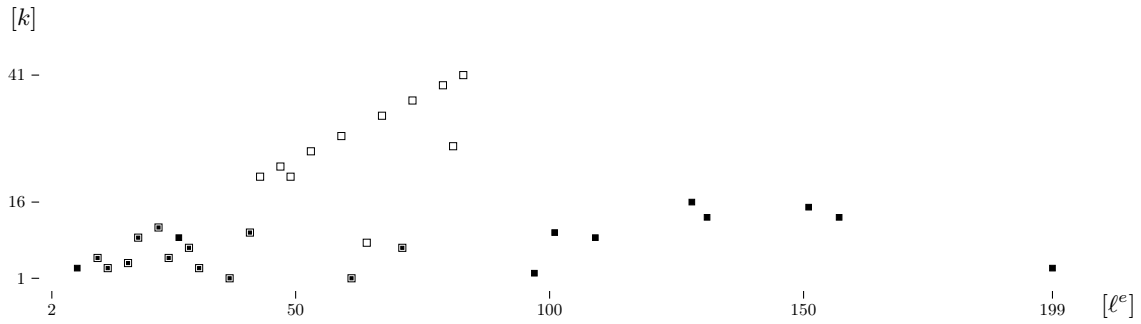


Fig. 2. Illustration of ℓ^e-torsion subgroups with their associated extension degree k chosen by naïve powersmooth KLPT (□) and our algorithm (■), for a quadratic-time cost model and some particular combination of prime p and magnitude of norm of the output ideal.

4.3 Irrational x-only arithmetic

A standard technique in elliptic-curve and isogeny-based cryptography is to work with x-coordinates only, instead of "full" points (x, y) having both coordinates. This allows us to compute both on the curve and its twists using unified formulas: x-coordinates for which the associated y-coordinate is irrational define rational points on the quadratic twist.

One important advantage is that by Theorem 2, if we start with a curve with $(p^k - (-1)^k)^2$ points, its twist has $(p^k + (-1)^k)^2$ points, allowing us to access more torsion. The special case $k = 1$ amounts to working with (p^2-1)-torsion, which is nowadays standard in isogeny-based cryptography [Cos20], but higher extensions are seldomly used.

Some computations can be performed entirely in x-only arithmetic: Using the fact that $x(-P) = x(P)$, it is easy to see that the x-coordinate of a scalar multiple $[n]P$ depends only on the x-coordinate of P. Therefore, for any $n \in \mathbb{Z}$ not divisible by the order of P, the map $x(P) \mapsto x([n]P)$ is well-defined, and it can be computed efficiently using only $O(\log |n|)$ operations in the base ring using a *ladder* algorithm. We write $\texttt{xMUL}(E, \xi, n)$ for such an algorithm, taking a curve $E/\mathbb{F}_q$, an x-coordinate ξ of a point on E, and a scalar $n \in \mathbb{Z}$. Note that `xMUL` is algebraic; in particular, ξ may be an element of any algebra over $\mathbb{F}_q$.

For other computations, typically those involving point additions, relying only on x-coordinates can become difficult or inefficient or both. In our implementation, we employ full points (either on the original curve or on a rational model of a suitable quadratic twist) up until the evaluation of the endomorphism $\overline{\alpha}$ in Algorithm 2, then drop the y-coordinates and perform all remaining computations in an x-only manner.

Kernel polynomials from irrational points. We need to compute many isogenies with kernels whose points are only defined over extension fields. As such, computing kernel polynomials is a bottleneck in our algorithm. Algorithm 4 computes the kernel polynomial from an irrational point (represented by its x-coordinate ξ) faster than naïvely enumerating points in K and applying a product tree (see Section 2.3). We introduce the following terminology, a mild generalization of the concept of kernel polynomials:

Definition 15. *Let E be an elliptic curve over a field k and $f \in k[X]$ a monic squarefree polynomial. The* subgroup defined by *f is the subgroup G of E generated by the set of points $\{P \in E\backslash\{0\} : f(x(P)) = 0\}$.*

In this situation, we say that f is a defining polynomial for G, and if f is furthermore irreducible, we refer to f as a minimal polynomial *of G.*

Note that every cyclic subgroup can be defined by a minimal polynomial; taking the minimal polynomial of the x-coordinate of any generating point of the subgroup suffices. Representing subgroups by their minimal polynomials instead of "full" kernel polynomials can save time, especially if the kernel points are defined over field extensions of degree much smaller than the isogeny degree — i.e., the particular scenario we are enforcing in our implementation. However, it does raise the algorithmic question of how to compute and evaluate isogenies when subgroups are represented by minimal polynomials. Answers will be given in this section.

Historical note. An algorithm very similar to Algorithm 3 was given in [Tsu13, § 3.4], but as described there it involves computing a greatest common divisor with the ℓ-division polynomial, which renders it less efficient than Algorithm 3. There is also no discussion of the complexity.

SageMath currently uses the algorithm from [Tsu13] to enumerate all ℓ-isogenies from a given curve. The implementation additionally features an efficiency improvement due to Demeyer [Dem15] that is mathematically identical to the technique used in Algorithm 3; however, it seems to have gone unnoticed that Shoup's algorithm is faster than the algorithm implemented in Sage (as of version 9.7).

Algorithm 3: `KernelPolynomialFromDivisor`(E, f, ℓ)

Input: Elliptic curve $E/\mathbb{F}_q$, prime integer ℓ, minimal polynomial $f \in \mathbb{F}_q[X]$ of an order-ℓ subgroup $G \leq E$.

Output: The kernel polynomial $h \in \mathbb{F}_q[X]$ of G.

Set $k \leftarrow \deg f$ and $m \leftarrow \lfloor \ell/2k \rfloor$ and $f_1 \leftarrow f$.
Search for a primitive root $a \in \mathbb{Z}$ modulo ℓ of minimal absolute value.
For i **from** 2 **to** m **do**
    Write $\overline{X}$ for the image of X in $\mathbb{F}_q[X]/f_{i-1}$ and compute $\alpha_i \leftarrow$ `xMUL`$(E, \overline{X}, a) \in \mathbb{F}_q[X]/f_{i-1}$.
    Find the minimal polynomial $f_i \in \mathbb{F}_q[X]$ of α_i over $\mathbb{F}_q$ using Shoup's algorithm.
Compute $h \leftarrow \prod_{i=1}^{m} f_i \in \mathbb{F}_q[X]$ using a product tree.
Return h.

Algorithm 4: `KernelPolynomialFromIrrationalX`(E, ξ, ℓ)

Input: Elliptic curve $E/\mathbb{F}_q$, extension $\mathbb{F}_{q^r}/\mathbb{F}_q$, x-coordinate $\xi \in \mathbb{F}_{q^r}$ of an order-ℓ point $P \in E$ lying in an eigenspace of the q-power Frobenius on E.

Output: The kernel polynomial $h_{\langle P \rangle} \in \mathbb{F}_q[X]$ defining the subgroup of E generated by P.

Find the minimal polynomial $\mu \in \mathbb{F}_q[X]$ of ξ over $\mathbb{F}_q$ using Shoup's algorithm.
Return `KernelPolynomialFromDivisor`(E, μ, ℓ).

Lemma 16. *Algorithm 3 is correct. It can be implemented in such a way that it runs within $O(\ell k) + \widetilde{O}(\ell)$ operations in the field $\mathbb{F}_q$.*

Proof. If $\ell = 2$, we must have $k = 1$ and $m = 1$ and the algorithm simply returns f. Assume $\ell \geq 3$ below.

Consider an arbitrary root $\xi \in \mathbb{F}_{q^k}$ of f and a point $P \in E$ with x-coordinate ξ. Let $Q = \pi(P)$ where $\pi\colon E \to E$ is the q-power Frobenius. Since f has coefficients in $\mathbb{F}_q$, the x-coordinate $x(Q) = \xi^q \in \mathbb{F}_{q^k}$ must be a root of f as well. By assumption the roots of f define points contained in a prime-order (hence cyclic) subgroup; thus, the associated points are linearly dependent: There exists $\lambda \in \mathbb{Z}$ such that $Q = [\lambda]P$. Iterated applications of π reveal that $f = \prod_{s \in S} \big(X - x([s]P)\big)$, where $S = \{1, \lambda, \lambda^2, ..., \lambda^{k-1}\}$.

The objective is to recover the "missing" factors of the kernel polynomial $h = \prod_{n=1}^{(\ell-1)/2} \big(X - x([n]P)\big)$. Since a is a primitive root modulo ℓ, it generates the quotient group $(\mathbb{Z}/\ell)^\times/S$, yielding the partition $(\mathbb{Z}/\ell)^\times = \bigcup_{i=0}^{(\ell-1)/k} a^i S = \pm \bigcup_{i=0}^{(\ell-1)/2k} a^i S$. Therefore, it suffices to show that the algorithm computes each $f_i = \prod_{s \in S} \big(X - x([a^{i-1}s]P)\big)$ correctly: The evaluation map $\mathbb{F}_q[X] \to \mathbb{F}_{q^k}$, $X \mapsto x([a^{i-2}]P)$ descends to an $\mathbb{F}_q$-algebra isomorphism $\iota\colon \mathbb{F}_q[X]/f_{i-1} \cong \mathbb{F}_{q^k}$. By definition of `xMUL`, we get $\iota(\alpha_i) = x([a^{i-1}]P)$, whose minimal polynomial over $\mathbb{F}_q$ equals f_i by the same reasoning as for f above.

Regarding the complexity: The loop runs $O(\ell/k)$ times. FFT-based arithmetic in $\mathbb{F}_q[X]/f_{i-1}$ requires $\widetilde{O}(k)$ operations in $\mathbb{F}_q$. Shoup's algorithm [Sho99] requires $O(k^2)$ operations in $\mathbb{F}_q$. The final product tree can be computed in $\widetilde{O}(\ell)$ operations in $\mathbb{F}_q$, again using FFT-based polynomial arithmetic. Simplify using $\ell/k \cdot \widetilde{O}(k) = \ell(\log k)^{O(1)}$ and $k, |a| \in O(\ell)$ to get the claimed runtime. □

Remark 17. The complexity of Algorithm 4 is $O(r^2)$ for Shoup's algorithm plus the time required by Algorithm 3. Hence, unless the given x-coordinate ξ is represented as an element of an excessively large extension field of degree $\notin O(\deg \mu)$, Algorithm 4 runs in time $O(\ell r) + \widetilde{O}(\ell)$ as well.

By comparison, the complexity of the straightforward algorithm outlined in Section 2.3 is $\widetilde{O}(\ell r)$.

Remark 18. Algorithm 3 has been restricted to irreducible f and prime orders ℓ for simplicity and ease of notation. It can be generalized to arbitrary orders ℓ, and reducible defining polynomials of G, assuming one is willing to deal with the added complications in the structure of the monoid $(\mathbb{Z}/\ell, \cdot)$ when ℓ is composite. However, the particular case of prime powers is fairly manageable and very useful.

Pushing subgroups through isogenies. This section presents an algorithm for finding the image of a finite subgroup under an isogeny when using the minimal-polynomial representation (Definition 15). The technique generalizes Steps 4 and 5 of Algorithm 3.

Algorithm 5: `PushSubgroup`(E, f, φ)

Input: Elliptic curve $E/\mathbb{F}_q$, minimal polynomial $f \in \mathbb{F}_q[X]$ of a subgroup $G \leq E$, isogeny $\varphi\colon E \to E'$ defined over $\mathbb{F}_q$.

Output: Minimal polynomial $f^\varphi \in \mathbb{F}_q[X]$ of the subgroup $\varphi(G) \leq E'$.

1 Write the x-coordinate map of φ as a fraction g_1/g_2 of polynomials $g_1, g_2 \in \mathbb{F}_q[X]$.
2 Let $g_{\ker} \leftarrow \gcd(g_2, f)$ and $f_1 \leftarrow f/g_{\ker}$.
3 Compute $g_1 \cdot g_2^{-1} \bmod f_1 \in \mathbb{F}_q[X]$ and reinterpret the result as a quotient-ring element $\alpha \in \mathbb{F}_q[X]/f_1$.
4 Find the minimal polynomial $f^\varphi \in \mathbb{F}_q[X]$ of α over $\mathbb{F}_q$ using Shoup's algorithm.
5 Return f^φ.

Lemma 19. *Algorithm 5 is correct. It can be implemented in such a way that it runs within $O(k^2)+\widetilde{O}(n)$ operations in the field $\mathbb{F}_q$, where $k = \deg f$ and $n = \deg \varphi$.*

Proof. Consider an arbitrary root $\xi \in \overline{\mathbb{F}_q}$ of f and a point $P \in E$ with x-coordinate ξ. If $g_{\ker}(\xi) = 0$, then $g_2(\xi) = 0$, so $P \in \ker \varphi$. Otherwise, we have $f_1(\xi) = 0$ and hence (similarly to the proof of Lemma 16) we get $\iota_1(\alpha) = g_1(\xi)/g_2(\xi) = x(\varphi(P))$ where $\iota_1\colon \mathbb{F}_q[X] \to \overline{\mathbb{F}_q}$, $X \mapsto \xi$. This shows that f^φ is the minimal polynomial of $x(\varphi(P))$. By Definition 15, the collection of all P considered here is a generating set of G, hence f^φ is a generating polynomial for the group generated by all the $\varphi(P)$.

Regarding the complexity: FFT-based arithmetic on polynomials of degree bounded by d takes $\widetilde{O}(d)$ operations in $\mathbb{F}_q$. Shoup's algorithm [Sho99] requires $O(k^2)$ operations in $\mathbb{F}_q$. The overall cost is therefore $\widetilde{O}(\max\{n, k\}) + O(k^2) \subseteq O(k^2) + \widetilde{O}(n)$. □

For comparison, the complexity of straightforward evaluation of φ at a single generating point of G with coordinates in $\mathbb{F}_{q^k}$ uses $O(nk(\log k)^{O(1)})$ operations in $\mathbb{F}_q$.

Remark 20. Algorithm 5 did not make any assumption on the degree of φ, but of course the reasonable thing to do in most cases will be to apply the algorithm to each prime-degree step of φ sequentially.

If φ is a scalar multiplication, it is better dealt with by running `xMUL` in the quotient ring $\mathbb{F}_q[X]/f$, as in Step 5 of Algorithm 3, rather than writing out the rational maps first.

5 Numerical examples and experiments

We start in Section 5.1 with a few examples illustrating the tools developed in Section 4: in Example 21 we give a worked example of the entire computation for a prime $p \equiv 1 \pmod{12}$, and in Example 22 we give an example of finding an isogeny connecting elliptic curves with j-invariants 1728 and 0.

We report on the timings of our algorithm in Section 5.2.

5.1 Examples

We begin by providing a numerical example to illustrate the algorithm.

Example 21. In this example we are working with $p = 61057$, a 16-bit prime with $p \equiv 1 \pmod{12}$. The quaternion algebra $B_{p,\infty}$ can be given the $\mathbb{Q}$-basis $\{1, \mathbf{i}, \mathbf{j}, \mathbf{k}\}$, where $\mathbf{i}^2 = -7$, $\mathbf{j}^2 = -p$ and $\mathbf{ij} = -\mathbf{ji} = \mathbf{k}$. In $B_{p,\infty}$, the order

$$\mathcal{O} = \mathbb{Z} \oplus \mathbb{Z}\,\frac{1 + 79\mathbf{i}}{2} \oplus \mathbb{Z}\,(3\mathbf{i} + \mathbf{j}) \oplus \mathbb{Z}\,\frac{553 + 35467\mathbf{i} + 987\mathbf{j} + \mathbf{k}}{1106}$$

is maximal, and our goal will be to construct an elliptic curve with endomorphism ring isomorphic to $\mathcal{O}$.

Step 0. The unique root of

$$H_{-7}(X) = X + 3375$$

in $\mathbb{F}_p$ is 57682, and we find that the curve $E_0/\mathbb{F}_p\colon y^2 = x^3 + 19621x + 41436$ has $j(E_0) = 57682$. Further, we find that the endomorphism ring of E_0 is isomorphic to

$$\mathcal{O}_0 = \mathbb{Z} \oplus \mathbb{Z}\,\frac{1 + \mathbf{i}}{2} \oplus \mathbb{Z}\,\frac{\mathbf{j} + \mathbf{k}}{2} \oplus \mathbb{Z}\,\frac{2\mathbf{i} - \mathbf{k}}{7},$$

where the endomorphism corresponding to $\mathbf{i}$ has kernel with a minimal polynomial (Definition 15)

$$X^3 + 30526X^2 + 23984X + 12309.$$

In order to find a curve with endomorphism ring isomorphic to $\mathcal{O}$, we need a connecting $(\mathcal{O}_0, \mathcal{O})$-ideal. We take the standard choice

$$I \;=\; N\mathcal{O}_0\mathcal{O} \;=\; \mathbb{Z}\,79 \oplus \mathbb{Z}\,\frac{79 + 79\mathbf{i}}{2} \oplus \mathbb{Z}\,(37 + 3\mathbf{i} + \mathbf{j}) \oplus \mathbb{Z}\,\frac{791 + 453\mathbf{i} + 7\mathbf{j} + \mathbf{k}}{14}$$

and aim to translate this ideal to its corresponding isogeny.

Step 1. We start by finding the target norm for the KLPT algorithm. We will do the simple translation in Step 2, hence we need $R > p^3$. We find that

$$\begin{array}{lllll} 2^{10} \mid p^8 - 1, & 3^4 \mid p^9 - 1, & 5^3 \mid p^2 + 1, & 7 \mid p^3 + 1, & 11 \mid p^5 + 1, \\ 13 \mid p^3 - 1, & 17 \mid p^8 + 1, & 19 \mid p^9 + 1, & 29 \mid p^2 + 1, & 53 \mid p - 1, \end{array}$$

and that $R = 2^{10} \cdot 3^4 \cdot 5^3 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 29 \cdot 53 > 227617885752193 = p^3$. Running KLPT, with R as a target norm, we find an equivalent ideal $J \sim I$ with $\mathrm{nrd}(J) \mid R$ as

$$\begin{aligned} J = \mathbb{Z}\,&92006270928000 \\ &\oplus \mathbb{Z}\,(31627155631500 + 2875195966500\mathbf{i}) \\ &\oplus \mathbb{Z}\,\frac{25167369945337 + 690338525003\mathbf{i} + 32\mathbf{j}}{2} \\ &\oplus \mathbb{Z}\,\frac{740914458532283 + 24241082699825\mathbf{i} + 21\mathbf{j} + \mathbf{k}}{14} \end{aligned}$$

with $\mathrm{nrd}(J) = 2^7 \cdot 3^4 \cdot 5^3 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 29 \cdot 53$.

Step 2. Next, we find generators for the full torsion groups:

$$\begin{array}{llll} E_0[2^8] \subseteq E_0(\mathbb{F}_{p^4}), & E_0[3^4] \subseteq \widetilde{E_0}(\mathbb{F}_{p^{18}}), & E_0[5^3] \subseteq \widetilde{E_0}(\mathbb{F}_{p^4}), & E_0[13] \subseteq \widetilde{E_0}(\mathbb{F}_{p^6}), \\ E_0[17] \subseteq \widetilde{E_0}(\mathbb{F}_{p^{16}}), & E_0[19] \subseteq E_0(\mathbb{F}_{p^{18}}), & E_0[29] \subseteq \widetilde{E_0}(\mathbb{F}_{p^4}), & E_0[53] \subseteq \widetilde{E_0}(\mathbb{F}_{p^2}). \end{array}$$

Here, $\widetilde{E}(\mathbb{F}_q)$ denote the $\mathbb{F}_q$-rational subgroup of a quadratic twist of E over $\mathbb{F}_q$.

Next, we determine the action of I on the different torsion groups (see Algorithm 2), and find a generator P_N for every (maximal) prime power $N \mid \mathrm{nrd}(J)$. Some of these generators are a priori only defined over twists, but using the twisting isomorphism we can transfer the x-coordinate back to E_0.

From the x-coordinates of these generators, we compute the corresponding chain of isogenies (using Algorithm 4) and end up at the curve

$$E_I/\mathbb{F}_p(\alpha)\colon\ y^2 = x^3 + (38455\alpha + 40273)x + (3066\alpha + 17732),$$

where α satisfies $\alpha^2 + 5 = 0$. From the Deuring correspondence, we know that

$$\mathrm{End}(E_I) \cong \mathcal{O}_R(J) \cong \mathcal{O}_R(I) = \mathcal{O}.$$

Example 22 (Connecting $j = 1728$ with $j = 0$). When $p \equiv 11 \pmod{12}$, both the curves $E_0\colon y^2 = x^3 + x$ and $E_1\colon y^2 = x^3 + 1$, of j-invariants 1728 and 0 respectively, are supersingular. This has prompted [CPV20, Example 19] to investigate the problem of finding an $\mathbb{F}_p$-rational isogeny between those two curves in the setting of the cryptographic group action CSIDH [CLMPR18]. In the following, we will arbitrarily work in characteristic $p = 7799999$ for the sake of an example and consider the analogous problem of finding an arbitrary, not necessarily $\mathbb{F}_p$-rational isogeny.

Recall from Section 3.1 that E has endomorphism ring $\mathcal{O}_0 = \langle 1, \mathbf{i}, (\mathbf{i}+\mathbf{j})/2, (1+\mathbf{k})/2\rangle$ with $\mathbf{i}^2 = -1$ while E' has endomorphism ring $\mathcal{O}_1' = \langle 1, (1+\mathbf{i}')/2, (\mathbf{j}'+\mathbf{k}')/2, (\mathbf{i}'+\mathbf{k}')/3\rangle$ with $\mathbf{i}'^2 = -3$.

Our first task is to map the order $\mathcal{O}_1'$ from the quaternion algebra $B' = (-3, -p)$ to the (isomorphic) quaternion algebra $B = (-1, -p)$, as to recover the embedding $\mathcal{O}_1' \hookrightarrow \mathcal{O}_0 \otimes \mathbb{Q}$ from Remark 7. To apply Lemma 10, we solve the Diophantine equation $x^2 + py^2 = 3$ over the rationals. The solution with the smallest denominator is $(x, y) = (598/1649, 1/1649)$, so we let $\gamma = (598 + \mathbf{j})/1649 \in B$.

Pulling back $\mathcal{O}_1'$ to $B = \mathcal{O}_0 \otimes \mathbb{Q}$ through the isomorphism from Lemma 10 gives the isomorphic order

$$\mathcal{O}_1 = \mathbb{Z} \oplus \mathbb{Z}\,4947\mathbf{i} \oplus \mathbb{Z}\,\frac{4947\mathbf{i} + \mathbf{j}}{2} \oplus \mathbb{Z}\,\frac{4947 + 32631010\mathbf{i} + \mathbf{k}}{9894}.$$

Hence, the connecting ideal $I = N\mathcal{O}_0\mathcal{O}_1$, where N smallest integer such that the ideal is integral, equals

$$I = N\mathcal{O}_0\mathcal{O}_1 = \mathbb{Z}\,4947 \oplus \mathbb{Z}\,4947\mathbf{i} \oplus \mathbb{Z}\,\frac{598 + 4947\mathbf{i} + \mathbf{j}}{2} \oplus \mathbb{Z}\,\frac{4947 + 598\mathbf{i} + \mathbf{k}}{2},$$

and its norm 4947 factors as $3 \cdot 17 \cdot 97$. Using (for instance) Theorem 2, we see that $E_0[3] \subseteq E_0(\mathbb{F}_{p^2})$, $E_0[17] \subseteq \widetilde{E_0}(\mathbb{F}_{p^8})$, and $E_0[97] \subseteq \widetilde{E_0}(\mathbb{F}_{p^6})$, where as before $\widetilde{E}(\mathbb{F}_q)$ denotes the $\mathbb{F}_q$-rational subgroup of a quadratic twist of E over $\mathbb{F}_q$.

With the explicit endomorphisms from Section 3.1, in particular $\iota\colon E_0 \to E_0, (x, y) \mapsto (-x, iy)$ where i is a fixed square root of -1 in $\mathbb{F}_{p^2}$, we may then run Algorithm 2 to compute generators of the subgroup of E_0 defined by I, and find the minimal polynomials of the x-coordinates to recover minimal polynomials of the kernel subgroups (Definition 15). One possible set of such minimal polynomials is:

$$\begin{aligned}
f_3 &= X + 1584399\,;\\
f_{17} &= X^4 + (1991643 + 7147424i)X^3 + (5285403 + 5254148i)X^2\\
&\qquad + (1481864 + 4554701i)X + (6263369 + 6535494i)\,;\\
f_{97} &= X^3 + (5961087 + 1392356i)X^2 + (7797495 + 394298i)X + (4229973 + 3176957i)\,.
\end{aligned}$$

The rest of the computation is done using Algorithms 3 and 5: We obtain the sequence of isogenies

$$E_0 \xrightarrow{\varphi_3} E' \xrightarrow{\varphi_{17}} E'' \xrightarrow{\varphi_{97}} E_1$$

where $E'\colon y^2 = x^3 + 808882x + 347859$ and $E''\colon y^2 = x^3 + 1607537x + 7524091$, and $\deg\varphi_d = d$. The images of the subgroups defined by f_{17} and f_{97} on E' and E'' are defined by the minimal polynomials

$$\begin{aligned}
f_{17}' &= X^4 + (5419201 + 308473i)X^3 + (940694 + 1289266i)X^2\\
&\qquad + (4123481 + 25574i)X + (5711471 + 1208667i)\,;\\
f_{97}' &= X^3 + (948701 + 1793351i)X^2 + (160774 + 5202674i)X + (5191824 + 6173732i)\,;\\
f_{97}'' &= X^3 + (3261011 + 405855i)X^2 + (6008102 + 1767374i)X + (460134 + 2885906i)\,.
\end{aligned}$$

5.2 Experiments

We implemented our algorithm in SageMath [The22], making use of its good library support for elliptic curves and isogenies over finite fields as well as quaternion algebras. It seems likely that one could obtain handsome practical speedups by switching to a lower-level programming language. Our hope is that our Sage implementation will be easy to use and extend for computational number theorists and isogenistas. The code is available at `https://github.com/friends-of-quaternions/deuring`.

Comparison to previous work. Earlier work by Kambe, Yasuda, Noro, Yokoyama, Aikawa, Takashima, and Kudo [Kam+22] deals with the Deuring correspondence for generic primes $p \equiv 3 \pmod 4$. In Step 1, they select R to be the smallest powersmooth number exceeding p^3. In Step 2 of the algorithm, they apply precomputed symbolic formulae for isogenies, to recover a factor of the ℓ-th division polynomial, which in turn recovers an ℓ-torsion point, and use this technique to construct a basis for the ℓ-torsion group. This basis can then be lifted to a basis of the ℓ^e-torsion group. Such formulas are currently only available for $\ell \leq 131$.

We compare our timings against the results from [Ray18] and [Kam+22] in Table 1.

Table 1. Comparison of our implementation to [Ray18; Kam+22].

Bit length	[Ray18, Figure 4.1]	[Kam+22, Table 4]	This work
9 ($p = 431$)	407 s	-	1.2s
11 ($p = 1619$)	718 s	-	1.46 s
15	-	45 s	1.333 s
20	-	447 s	1.281 s
25	-	392 s	1.792 s

Generic primes. We tested our implementation on random primes between 5 and 255 bits, in steps of 5 bits. The results are summarized in Figure 3. The target orders were chosen by sampling representatives I of random left-ideal classes of $\mathcal{O}_0$, after $\mathcal{O}_0$ was constructed as described in Section 3.1. Each I was sampled by growing a chain of $55 + \lceil \log_2 p \rceil$ connecting norm-2 ideals starting from $\mathcal{O}_0$ uniformly at random, intermittently replacing the chain by an equivalent ideal of smaller norm when the integers used to write the basis elements grew too big.

To evaluate the effectiveness of our torsion-selection optimization from Section 4.2, we also compare to a "naïve" variant of our implementation, in which the KLPT phase does not pay any attention to the field extension degrees and instead simply selects prime-power torsion subgroups in ascending order as suggested by prior literature [GPS17; EHLMP18].

Nice primes. The runtime can vary a lot for primes of similar size. By carefully choosing p, it is possible to construct exceptional cases, where the computation of the Deuring correspondence becomes much faster than average. For instance, In SQISign, the prime p is selected such that $p^2 - 1$ factors favorably, allowing $E[R] \subseteq E(\mathbb{F}_{p^4})$ (note that they use a much smaller R than our implementation, see Appendix A).

We also test our algorithm against primes specifically constructed to facilitate the computation of the Deuring correspondence. Specifically, we run the computation on three different primes of ≈ 256 bits:

- p_{3923}: The 254-bit prime currently used in SQISign [DLW22],
- p_1: A 253-bit prime specifically constructed for our implementation, using techniques from [Bru+22],
- p_2: A 255-bit prime from [Bru+22], suggested for instantiating SQISign,

Table 2 shows the results of these experiments. They demonstrate how our approach "automatically" benefits from the particular structure of the carefully selected primes: For all these primes, the runtime is about one order of magnitude faster than a random 255-bit prime (see Figure 3).

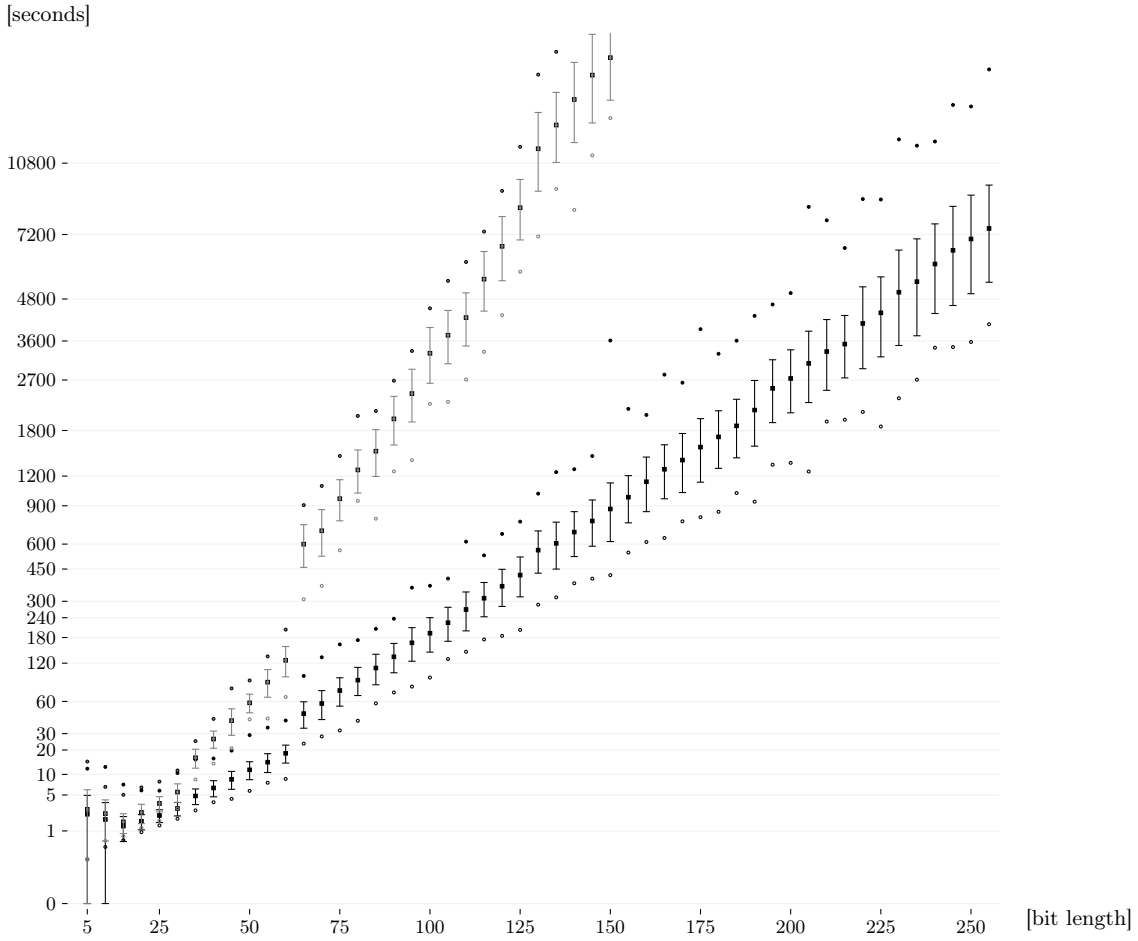


Fig. 3. Timings for running our implementation of the Deuring correspondence for random primes up to 255 bits (in steps of 5 bits), showing mean (■), minimum (○), maximum (•), and estimated standard deviation (error bars). The y-axis uses a quartic scale. Each data point represents measurements from 256 independent runs. Experiments were run in parallel, one instance per core at a time, using SageMath 9.7 on a server at Academia Sinica with two 64-core AMD EPYC 7763 processors.
Plotted in gray are runtime measurements when the cost model simply picks powersmooth torsion subgroups of increasing size while ignoring field extension degrees. These timings are from 48 separate runs up to 150 bits on two servers with a total of four 12-core Intel Skylake processors (Xeon Gold 5118, Xeon Gold 6136).
Note that the speeds for very small bit lengths can be beaten with a simple-minded brute-force approach which does not rely on KLPT at all.

We also see that our algorithm runs fastest for p_1. While p_2 and p_{3923} was constructed with SQISign in mind (hence only looking for ℓ^e such that the multiplicative order of p in $\mathbb{Z}/\ell^e\mathbb{Z}$ is at most 2), the prime p_1 was constructed allowing slightly higher multiplicative orders of p. The integer values for these primes can be found in Appendix B.

Remark 23 (Comparison with SQISign). The SQISign implementation will undoubtedly run the computation of the Deuring correspondence for p_2 and p_{3923} much faster, but it is not really a fair comparison: By virtue of working with fixed primes, the SQISign implementation can precompute the actions of the generators of $\mathcal{O}_0$ on fixed torsion bases on E_0, which are heavy computations that our generic implementation has to perform on the fly.

Table 2. The results of running our implementation on 3 primes specifically chosen to facilitate the computation of the Deuring correspondence. The computations were run using SageMath 9.7 on a laptop with an Intel Core i5-1038NG7 processor.

Prime	Time	Top 3 most expensive torsion groups to work with (per cost model)
p_{3923}	1863 s	$E[4733] \subseteq E(\mathbb{F}_{p^{52}}), E[3^{68}] \subseteq E(\mathbb{F}_{p^{54}}), E[109] \subseteq E(\mathbb{F}_{p^{54}})$
p_1	962 s	$E[13789] \subseteq E(\mathbb{F}_{p^{36}}), E[691] \subseteq E(\mathbb{F}_{p^{46}}), E[461] \subseteq E(\mathbb{F}_{p^{46}})$
p_2	1578 s	$E[409] \subseteq E(\mathbb{F}_{p^{68}}), E[1321] \subseteq E(\mathbb{F}_{p^{66}}), E[859] \subseteq E(\mathbb{F}_{p^{66}})$

References

[BCEMP19] Efrat Bank, Catalina Camacho-Navarro, Kirsten Eisenträger, Travis Morrison, and Jennifer Park. "Cycles in the Supersingular ℓ-Isogeny Graph and Corresponding Endomorphisms". In: *Research Directions in Number Theory*. Ed. by Jennifer S. Balakrishnan, Amanda Folsom, Matilde Lalín, and Michelle Manes. Springer, 2019, pp. 41–66. ISBN: 978-3-030-19478-9.

[BDLS20] Daniel J. Bernstein, Luca De Feo, Antonin Leroux, and Benjamin Smith. "Faster computation of isogenies of large prime degree". In: *ANTS XIV: Proceedings of the fourteenth algorithmic number theory symposium*. Ed. by Steven Galbraith. Auckland: Mathematical Sciences Publishers, 2020, pp. 39–55. URL: https://iac.r/2020/341.

[BMSS08] Alin Bostan, François Morain, Bruno Salvy, and Éric Schost. "Fast algorithms for computing isogenies between elliptic curves". In: *Mathematics of Computation* 77.263 (2008), pp. 1755–1778. URL: https://arxiv.org/abs/cs/0609020.

[Boo+22] Jeremy Booher, Ross Bowden, Javad Doliskani, Tako Boris Fouotsa, Steven D. Galbraith, Sabrina Kunzweiler, Simon-Philipp Merz, Christophe Petit, Benjamin Smith, Katherine E. Stange, Yan Bo Ti, Christelle Vincent, José Felipe Voloch, Charlotte Weitkämper, and Lukas Zobernig. *Failing to hash into supersingular isogeny graphs*. Cryptology ePrint Archive, Report 2022/518. 2022. URL: https://ia.cr/2022/518.

[Brö09] Reinier Bröker. "Constructing supersingular elliptic curves". In: *Journal of Combinatorics and Number Theory* 1.3 (2009), pp. 269–273.

[Bru+22] Giacomo Bruno, Maria Corte-Real Santos, Craig Costello, Jonathan Komada Eriksen, Michael Naehrig, Michael Meyer, and Bruno Sterner. *Cryptographic Smooth Neighbors*. Preprint. 2022. URL: https://ia.cr/2022/1439.

[Cer04] Juan Marcos Cerviño. *On the Correspondence between Supersingular Elliptic Curves and maximal quaternionic Orders*. Preprint. 2004. URL: https://arxiv.org/abs/math/0404538.

[CG14] Ilya Chevyrev and Steven D. Galbraith. "Constructing supersingular elliptic curves with a given endomorphism ring". In: *LMS Journal of Computation and Mathematics* 17.A (2014), pp. 71–91. DOI: 10.1112/S1461157014000254.

[CK91] David G. Cantor and Erich Kaltofen. "On Fast Multiplication of Polynomials over Arbitrary Algebras". In: *Acta Informatica* 28.7 (1991), pp. 693–701.

[CLMPR18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. "CSIDH: An Efficient Post-Quantum Commutative Group Action". In: Lecture Notes in Computer Science 11274 (2018), pp. 395–427. URL: https://ia.cr/2018/383.

[Cor08] Giuseppe Cornacchia. "Su di un metodo per la risoluzione in numeri interi dell' equazione $\sum_{h=0}^{n} C_h x^{n-h} y^h = P$". In: *Giornale di Matematiche di Battaglini* 46 (1908), pp. 33–90.

[Cos20] Craig Costello. "B-SIDH: Supersingular Isogeny Diffie-Hellman Using Twisted Torsion". In: *ASIACRYPT (2)*. Vol. 12492. Lecture Notes in Computer Science. Springer, 2020, pp. 440–463. URL: https://ia.cr/2019/1145.

[CPV20] Wouter Castryck, Lorenz Panny, and Frederik Vercauteren. "Rational Isogenies from Irrational Endomorphisms". In: *EUROCRYPT (2)*. Vol. 12106. Lecture Notes in Computer Science. Springer, 2020, pp. 523–548. URL: https://ia.cr/2019/1202.

[CX22] Mingjie Chen and Jiangwei Xue. *On $\mathbb{F}_p$-roots of the Hilbert class polynomial modulo p*. Preprint. 2022. URL: https://arxiv.org/abs/2202.04317.

[DeF+20] Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. "SQISign: Compact Post-quantum Signatures from Quaternions and Isogenies". In: *ASIACRYPT (1)*. Vol. 12491. Lecture Notes in Computer Science. Springer, 2020, pp. 64–93. URL: https://ia.cr/2020/1240.

[Dem15] Jeroen Demeyer. *Further isogeny improvement*. Ticket on the SageMath Developer Trac. 2015. URL: https://trac.sagemath.org/ticket/18611.

[Deu41] Max Deuring. "Die Typen der Multiplikatorenringe elliptischer Funktionenkörper". In: *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg* 14 (1941), pp. 197–272.

[DJP14] Luca De Feo, David Jao, and Jérôme Plût. "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies". In: *Journal of Mathematical Cryptology* 8.3 (2014), pp. 209–247. DOI: doi:10.1515/jmc-2012-0015. URL: https://ia.cr/2011/506.

[DLW22] Luca De Feo, Antonin Leroux, and Benjamin Wesolowski. *New algorithms for the Deuring correspondence: SQISign twice as fast*. Preprint. 2022. URL: https://ia.cr/2022/234.

[EHLMP18] Kirsten Eisenträger, Sean Hallgren, Kristin E. Lauter, Travis Morrison, and Christophe Petit. "Supersingular Isogeny Graphs and Endomorphism Rings: Reductions and Solutions". In: *EUROCRYPT (3)*. Vol. 10822. Lecture Notes in Computer Science. Springer, 2018, pp. 329–368. URL: https://ia.cr/2018/371.

[EHLMP20] Kirsten Eisentraeger, Sean Hallgren, Chris Leonardi, Travis Morrison, and Jennifer Park. "Computing endomorphism rings of supersingular elliptic curves and connections to pathfinding in isogeny graphs". In: *ANTS XIV: Proceedings of the fourteenth algorithmic number theory symposium*. Ed. by Steven Galbraith. Auckland: Mathematical Sciences Publishers, 2020, pp. 215–232. URL: https://arxiv.org/abs/2004.11495.

[Gal12] Steven D. Galbraith. *Mathematics of Public Key Cryptography*. Cambridge University Press, 2012. URL: https://www.math.auckland.ac.nz/~sgal018/crypto-book/crypto-book.html.

[GPS17] Steven D. Galbraith, Christophe Petit, and Javier Silva. "Identification Protocols and Signature Schemes Based on Supersingular Isogeny Problems". In: *ASIACRYPT (1)*. Vol. 10624. Lecture Notes in Computer Science. Springer, 2017, pp. 3–33. URL: https://ia.cr/2016/1154.

[Ibu82] Tomoyoshi Ibukiyama. "On maximal orders of division quaternion algebras over the rational number field with certain optimal embeddings". In: *Nagoya Mathematical Journal* 88 (1982), pp. 181–195.

[Kam+22] Yuta Kambe, Masaya Yasuda, Masayuki Noro, Kazuhiro Yokoyama, Yusuke Aikawa, Katsuyuki Takashima, and Momonari Kudo. "Solving the Constructive Deuring Correspondence via the Kohel–Lauter–Petit–Tignol Algorithm". In: *Mathematical Cryptology* 1.2 (2022), pp. 10–24. URL: https://journals.flvc.org/mathcryptology/article/view/130618.

[KLPT14] David Kohel, Kristin Lauter, Christophe Petit, and Jean-Pierre Tignol. "On the quaternion ℓ-isogeny path problem". In: *LMS Journal of Computation and Mathematics* 17 (2014), pp. 418–432. URL: https://ia.cr/2014/505.

[Koh96] David Kohel. "Endomorphism rings of elliptic curves over finite fields". PhD thesis. University of California at Berkeley, 1996. URL: https://www.i2m.univ-amu.fr/perso/david.kohel/pub/thesis.pdf.

[LB20] Jonathan Love and Dan Boneh. "Supersingular Curves With Small Non-integer Endomorphisms". In: *ANTS XIV: Proceedings of the fourteenth algorithmic number theory symposium*. Ed. by Steven Galbraith. Auckland: Mathematical Sciences Publishers, 2020, pp. 7–22. URL: https://arxiv.org/abs/1910.03180.

[Len96] Hendrik W. Lenstra. "Complex multiplication structure of elliptic curves". In: *Journal of Number Theory* 56 (2 1996), pp. 227–241.

[Ler22] Antonin Leroux. "Quaternion algebras and isogeny-based cryptography". PhD thesis. Ecole doctorale de l'Institut Polytechnique de Paris, 2022.

[McM14] Ken McMurdy. *Explicit representation of the endomorphism rings of supersingular elliptic curves*. Preprint. 2014. URL: https://phobos.ramapo.edu/~kmcmurdy/research/McMurdy-ssEndoRings.pdf.

[ML04] Ken McMurdy and Kristin Lauter. *Explicit Generators for Endomorphism Rings of Supersingular Elliptic Curves*. Preprint. 2004. URL: https://phobos.ramapo.edu/~kmcmurdy/research/ss_endomorphisms.pdf.

[PS18] Christophe Petit and Spike Smith. "An improvement to the quaternion analogue of the ℓ-isogeny path problem". In: *MathCrypt 2018*. 2018.

[Ray18] Dimitrij Ray. "Constructing the Deuring correspondence with applications to supersingular isogeny-based cryptography". Master's thesis. Technische Universiteit Eindhoven, 2018. URL: https://research.tue.nl/files/109549304/Dimitrij_Ray.pdf.

[Sch87] René Schoof. "Nonsingular plane cubic curves over finite fields". In: *Journal of Combinatorial Theory, Series A* 46.2 (1987), pp. 183–211.

[Sho99] Victor Shoup. "Efficient Computation of Minimal Polynomials in Algebraic Extensions of Finite Fields". In: *ISSAC '99*. ACM, 1999, pp. 53–58. DOI: 10.1145/309831.309859. URL: https://shoup.net/papers/mpol.pdf.

[Shu09] Daniel Shumow. "Isogenies of Elliptic Curves: A Computational Approach". Master's thesis. University of Washington, 2009. URL: https://sagemath.org/files/thesis/shumow-thesis-2009.pdf.

[Sil09] Joseph H. Silverman. *The Arithmetic of Elliptic Curves*. 2nd ed. Vol. 106. Graduate Texts in Mathematics. Springer, 2009. ISBN: 978-0-387-09493-9. DOI: 10.1007/978-0-387-09494-6.

[Sim05] Denis Simon. "Solving quadratic equations using reduced unimodular quadratic forms". In: *Mathematics of Computation* 74.251 (2005), pp. 1531–1543.

[The22] The Sage Developers. *SageMath, the Sage Mathematics Software System (version 9.7)*. https://sagemath.org. 2022.

[Tsu13] Kiminori Tsukazaki. "Explicit isogenies of elliptic curves". PhD thesis. University of Warwick, 2013. URL: https://wrap.warwick.ac.uk/57568.

[Vél71] Jacques Vélu. "Isogénies entre courbes elliptiques". In: *Comptes Rendus de l'Académie des Sciences de Paris*. A 273.4 (1971), pp. 238–241. URL: https://gallica.bnf.fr/ark:/12148/bpt6k56191248/f52.item.

[Voi21] John Voight. *Quaternion Algebras*. 2021. ISBN: 978-3-030-56692-0. DOI: 10.1007/978-3-030-56694-4. URL: https://math.dartmouth.edu/~jvoight/quat-book.pdf.

[Wat69] William C. Waterhouse. "Abelian varieties over finite fields". In: *Annales scientifiques de l'École Normale Supérieure* 2 (4 1969), pp. 521–560.

[Wes21] Benjamin Wesolowski. "The supersingular isogeny path and endomorphism ring problems are equivalent". In: *FOCS*. IEEE, 2021, pp. 1100–1111. URL: https://ia.cr/2021/919.

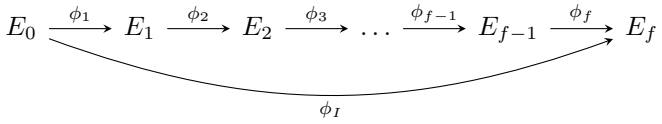
[Wes22] Benjamin Wesolowski. "Orientations and the Supersingular Endomorphism Ring Problem". In: *EUROCRYPT (3)*. Vol. 13277. Lecture Notes in Computer Science. Springer, 2022, pp. 345–371. URL: https://ia.cr/2021/1583.

A Isogeny Slide

In this Appendix, we explain an alternative way to compute the ideal-to-isogeny translation, based on [DeF+20]. The advantage is that it requires less torsion available. Our implementation, summarized in Algorithm 6, does not perform better than the direct ideal-to-isogeny translation following Section 3.3 and Section 4. However, we include it to complete our catalogue of known techniques for translating ideals to isogenies, and for future explorations of applicability.

This strategy is a simplified version of an idea originally described in SQISign: write $R = ST$, where S, T are coprime, $T > p^{3/2}$ and $S \ll T$. In practice, we search for $R > p^2$, select $T \mid R$ satisfying $T > p^{3/2}$ and taking $S = R/T \approx \sqrt{p}$. Once we have T and S, we run KLPT to find an equivalent ideal J with $n(J) \mid S^f$ such that $S^f \approx p^3$.

Translating the ideal to isogeny is then done as in Algorithm 6, which we describe next. The idea is to write the ideal I of norm diving S^f as the product $I = I_1 I_2 \cdots I_f$ of f integral ideals I_i of norm dividing S, and iteratively translate each I_i to an isogeny $\phi_i \colon E_{i-1} \to E_i$ separately, as in the following diagram:



The major issue is that the ideal-to-isogeny translation for some I_k requires that the endomorphism ring of E_{k-1} be *effective* in the sense of Section 2.6, see also Remark 7.

We can remedy the situation by *pulling back* the ideal I_k to the base curve E_0 through a different isogeny $\psi_{k-1} \colon E_0 \to E_{k-1}$ *of degree coprime to* S, computing the $\mathrm{nrd}(I_k)$-part of the kernel there, and pushing it back forward to E_{k-1} through ψ_{k-1}. Concretely, this amounts to the following steps: Replace $I_1 \cdots I_{k-1}$ with an equivalent ideal $J_k \sim I_1 \cdots I_{k-1}$ of norm coprime to S. Writing $J_k = I_1 \cdots I_{k-1}\beta$ with $\beta \in B_{p,\infty}^{\times}$, the product $J_k \beta^{-1} I_k \beta$ is a left $\mathcal{O}_0$-ideal equivalent to $I_1 \cdots I_k$ of norm $\mathrm{nrd}(J_k)\mathrm{nrd}(I_k)$; thus, the corresponding isogeny factors as $\phi_{I_k} \circ \phi_{J_k}$ (for appropriate choices of elliptic curves and endomorphism ring isomorphisms which respect the ideal decompositions). Compute the subgroup $K = E_0[\mathrm{nrd}(I_k)] \cap E_0[J_k I_k]$ defined by the *pullback ideal* $[J_k]^* I_k = \mathcal{O}_0 \mathrm{nrd}(I_k) + J_k I_k$ and push it through ϕ_{J_k} to obtain $\ker \phi_{I_k} = \phi_{J_k}(K)$. Repeating the same process for $k = 1, ..., f$ then translates the whole ideal I. This "sliding" procedure is summarized in Figure 4.

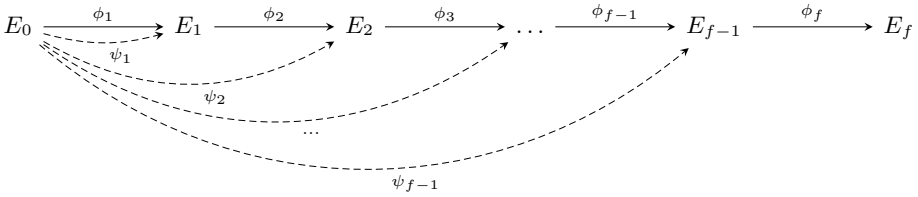


Fig. 4. An illustration of the `IdealToIsogenySlide` process, as in Algorithm 6.

The trick that makes Algorithm 6 require less rational torsion is the following *meet-in-the-middle* subroutine from SQISign [DeF+20], called `SpecialIdealToIsogeny`. It takes as input an ideal J of norm dividing T^2, and an ideal I of norm dividing S^f where T and S are coprime, and the isogeny ϕ_I. It outputs the isogeny ϕ_J. It works by writing as a product $J = J'J''$, both of norm dividing T. The ideal J' is a left O_0-ideal, so we can translate it directly. By using the ideal I of coprime norm, we can translate J'' by computing the kernel of the pullback ideal $[I]^*\overline{J''}$, and push it through ϕ_I to obtain $\ker \widehat{\phi_{\bar{J}''}} = \ker \widehat{\phi_{J''}}$. One can then recover ϕ_J as $\widehat{\phi_{\bar{J}''}} \circ \phi_{J'}$.

This approach requires translating many extra ideals to isogenies. However, since we need less available torsion, we can work with lower degree isogenies, and stay in lower degree extension fields. We tested Algorithm 6 against the same primes and ideals as in Section 5. We used different constants in the cost model, due to the fact that this technique requires many more isogeny translations, but generating torsion bases still only happens once. Specifically, we used $(c_1, c_2, c_3, c_4) = (0.31, 17.55, 6.90, 0.15)$. The results

Algorithm 6: `IdealToIsogenySlide`(I, E_0)

Input: left $\mathcal{O}_0$-ideal I of norm S^f, curve E_0 with effective endomorphism ring $\mathrm{End}(E_0) \cong \mathcal{O}_0$.
Output: ϕ_I.

Compute $I_1, \ldots, I_f$ such that $I = I_1 \cdots I_f$, and $\mathrm{nrd}(I_i) \mid S$ for all i.
Set $J_1 := \mathcal{O}_0$.
Set $\phi_{J_1} : E_0 \to E_0$ to be the identity on E_0.
For $i \in \{1, \ldots, f\}$ **do**
    Compute $[J_i]^* I_i$ as $J_i I_i + \mathcal{O}_0 n(I_i)$.
    Compute K as the group generated by `IdealToKernelGens`$([J_i]^* I_i, E_0)$.
    Compute ϕ_{I_i} as `Kohel`$(\phi_{J_i}(K))$.
    Compute $J_{i+1} \sim I_1 \cdots I_i$ with KLPT, where $n(J_{i+1}) \mid T^2$.
    Compute $\phi_{J_{i+1}}$ from `SpecialIdealToIsogeny`$(J_{i+1}, I_1 \cdots I_i, \phi_{I_i} \circ \cdots \circ \phi_{I_1})$.
Return $\phi_{I_f} \circ \cdots \circ \phi_{I_1}$.

are shown in Figure 5. The empirical data shows no sign of an asymptotic improvement in complexity compared to the direct ideal-to-isogeny translation, and preforms worse for us in practice. We do not rule out the possibility that this technique may become faster with different parameters and optimisations.

B Custom primes used in experiments

Here we give the custom primes used in Section 5, for reproducibility.

$$p_1 = 119565669446415029577041895949094989934782974038386434060581803341306567501 61$$
$$p_2 = 376705683365515363895039196659374911112161224703338376772138774424453119999 99$$
$$p_{3923} = 237593992641573523586737886133079705286468151140908767846433876621924499455 99$$

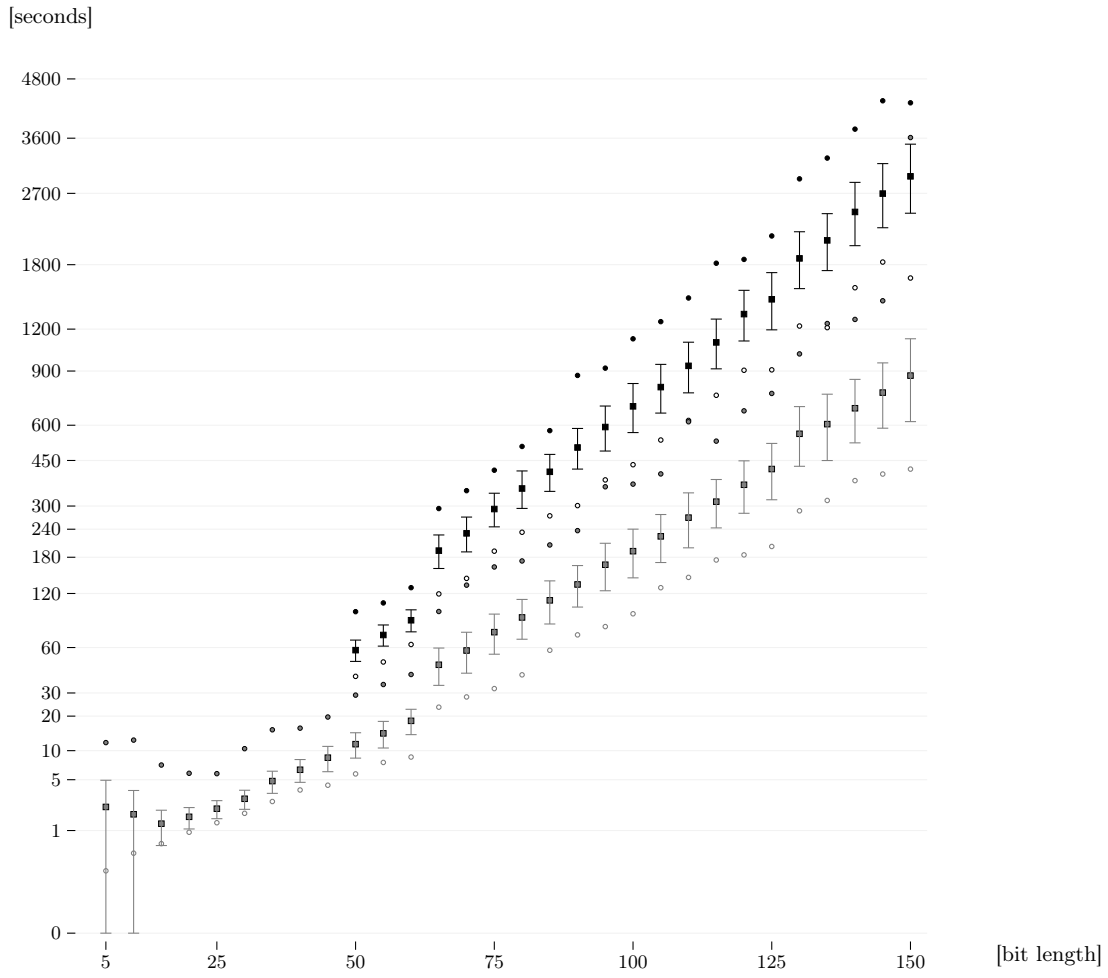


Fig. 5. Timings for running our implementation of the Deuring correspondence using Algorithm 6 for primes up to 150 bits, with the same experimental setup as for Figure 3. Each data point represents measurements from 128 independent runs. Data points from Figure 3 shown in gray for reference.