Jonathan Komada Eriksen

Supersingular Endomorphism Rings: Algorithms and Applications

Thesis for the Degree of Philosophiae Doctor

Trondheim, May 2024

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Information Security and Communication Technology

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Department of Information Security and Communication Technology

ISBN 978-82-326-8198-3 (printed version)
ISBN 978-82-326-8197-6 (electronic version)

Doctoral theses at NTNU

Printed by NTNU Grafisk senter

Preface

This thesis concludes my eight-year-long education in Trondheim, which consisted of a five-year masters degree, and a four-year employment as a PhD student[1] at the Department of Information Security and Communication Technology at the Norwegian University of Science and Technology.

The thesis consists of three parts, where the second and third part consists of three papers each. These papers are, for all intents and purposes, identical to the original papers, except for possible cosmetic changes to fit the format of this thesis. I would like to give a huge thanks all my co-authors for these fun projects, and for all you taught me.

In the first part of the thesis, I have attempted to give a thorough, but explicit account of the mathematics required for understanding these papers, and other algorithms related to the Deuring correspondence and isogeny-based cryptography. It contains a few deep, theoretical, and sometimes quite technical results; however, to bring things back to earth, I have attempted to accompany this with very thorough, and explicit examples. The examples follow the whole background section, and build on each other: The dependencies are given in Figure 1. I would also like to give an extra thanks to Colin Boyd, Craig Costello, Valerie Gilchrist and Anaëlle Le Dévéhat for having read through and suggested improvements in this part.

Finally, if you are more interested in poems than mathematics, I trust you will not be disappointed, as this thesis also contains three poems I am particularly proud of; they are some of my finest work.

[1] Indeed, $5 + 4 \neq 8$, the first of hopefully not too many mistakes in this thesis.

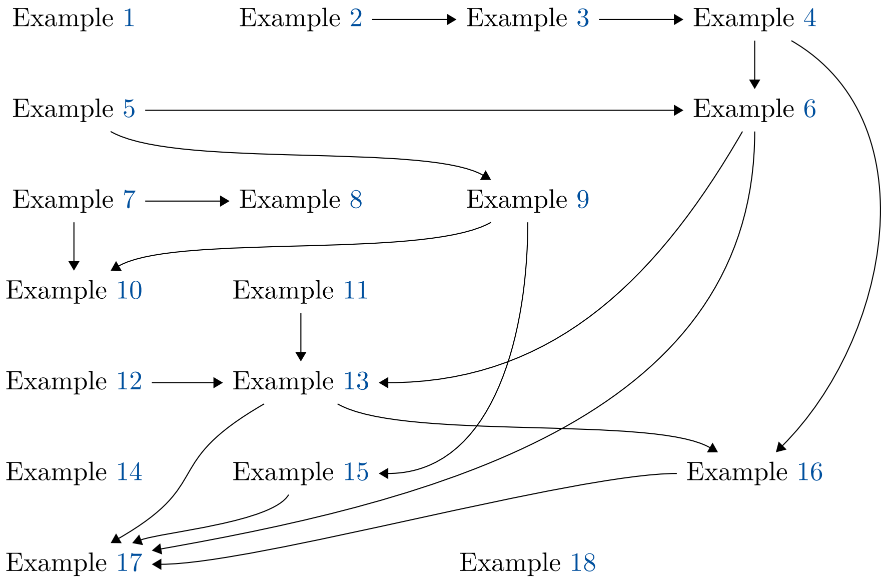


Figure 1: Dependencies among the examples, i.e. if $A \to B \to C$, you should really read both A and B, before reading C.

Acknowledgments

"Okei, jeg vi' ba' takke fammiljen, vennene, fænsen,... og [inaudible]*, OOOOOI!!!"*

Joachim "Jokke" Nielsen
Spellemannsprisen, 1991

To Colin: Thank you for your invaluable mentoring and guidence, both in research, and life in general.

To Craig: Thank you for being a true inspiration in my research, and for all of your genuine hype and motivation.

To Kristian: Thank you for introducing me to the joy of cryptography, which sent me tumbling down the rabbit hole.

To Mamma, Mike, Pappa and Lise: Thank you for everything.

To Miya, Kai and Alma: Thank you for making every trip home so great. You are all so talented in everything you are doing, there is no limit to what you can do.[2]

To Kristoffer: Thank you for always being my stand-in big brother.

To all of my friends in Trondheim: Thank you for always making each week more fun than the last.

To all my research friends around the world: Thank you for disproving all of the fears I had about research, all the good times we've had, and all the good times to come!

And finally, to all my friends in Oslo: Thank you for always being there for me, despite the geographical distance, for being the biggest constant in an otherwise volatile life and for being who you are.

Jonathan
Trondheim, May 2024

[2] Except beat me at any video game, ever, of course!

Sammendrag

Denne oppgaven omhandler endomorfi-ringene til supersingulære elliptiske kurver, applikasjoner av dette innen kryptografi, og relaterte algoritmer. Disse algoritmene og applikasjonene involverer alle et samspill mellom supersingulære elliptiske kurver og kvaternion algebraer, som er relatert gjennom endomorfi-ringene til de elliptiske kurvene.

Bidragene kan naturlig deles i to kategorier. De første bidragene omhandler effektive algoritmer for å oversette idealer i kvaternion algebraer til korresponderende isogenier, og kryptografiske applikasjoner av slike algoritmer. Spesifikt, så gir vi den første praktiske implementasjonen av denne oversettelsen fra idealer til isogenier som fungerer for *generelle* primtall, og videre ser vi på hvordan man kan finne *spesielle* primtall, slik at denne oppgaven blir enklere. Til sist anvender vi erfaringene fra disse bidragene til signatur algoritmen SQIsign. Dette gir en variant av SQIsign som har ekstra rask verifikasjon.

Det andre bidragene vi ser på er algoritmer, teori, og applikasjoner av optimale imbeddinger (imbeddinger av kvadratiske ordener i kvaternion ordener) og primitive orienteringer, som er optimale imbeddinger i endomorfiringene til supersingulære kurver. Det første bidraget her er en algoritme for å finne slike imbeddinger, som er asymptotisk raskere enn noen annen kjent algoritme for dette formålet. Fra denne algoritmen viser vi at det følger andre asymptotiske forbedringer til algoritmer relatert til supersingulære elliptiske kurver, og isogenier mellom disse. I det andre bidraget generaliserer vi teorien rundt primitive orienteringer, og viser at den velkjente klassegruppevirkningen på orienterte kurver er kun en liten del av et større bilde som inneholder generaliserte klassegrupper og Γ-nivå strukturer. Det siste bidraget er praktiske forbedringer til det kryptografiske primitivet SCALLOP, som er basert nettopp på denne velkjente klassegruppevirkningen. Ved å tillate litt mer generelle klassegruppestrukturer, får vi en betydelig raskere versjon av SCALLOP, som i tillegg kan instansieres ved høyere sikkerhetsnivå enn den originale versjonen.

Abstract

This thesis is about the endomorphism rings of supersingular elliptic curves, their applications in cryptography, and related algorithms. These algorithms and applications all involve an interplay between supersingular elliptic curves and quaternion algebras, which are related precisely through the endomorphism ring of the elliptic curves.

We divide the contributions in this thesis into two cases. The first is concerned with efficient algorithms for translating quaternion ideals to their corresponding isogenies, and their cryptographic applications. More specifically, we give the first practical implementation of this ideal to isogeny translation which works for *general* primes, and expand the literature on creating *special* primes that make the ideal to isogeny translation easier. Finally, we apply the lessons learned in these contributions to the signature scheme SQIsign. This gives a SQIsign-variant that has particularly fast verification.

The other case we consider is the algorithms, theory, and applications related to optimal embeddings (embeddings of quadratic orders into quaternion orders) and primitive orientations, which are optimal embeddings into the endomorphism rings of supersingular curves. The first contribution here is the asymptotically fastest algorithm for computing such embeddings. From this algorithm, we again derive other asymptotic improvements to algorithms for solving problems related to supersingular elliptic curves and their isogenies. For the second contribution, we generalise the theory of primitive orientations and show that the well-known class group action on oriented curves is a special case of a larger story involving generalised class groups and Γ-level structures. The final contributions we give are practical improvements to the cryptographic primitive SCALLOP, which is based on this well-known class group action. By relaxing certain requirements on the class group structure, we get a significantly faster version of SCALLOP, which is also possible to instantiate at higher security levels than the original version.

Contents

Introduction **1**

Outline and Summary of Contributions . . . 2

I Background **7**

1 Central Simple Algebras over the Rationals **9**

1.1 Preliminary Definitions . . . 9

1.2 Imaginary Quadratic Number Fields . . . 12

1.2.1 Imaginary Quadratic Orders and Their Ideals . . . 12

1.2.2 The Class Group . . . 15

1.2.3 Binary Quadratic Forms . . . 17

1.2.4 Computing in the Class Group . . . 19

1.3 Quaternion Algebras . . . 21

1.3.1 Orders in Quaternion Algebras, and Their Ideals . . . 24

1.3.2 The Class Set . . . 27

1.3.3 Ternary and Quartic Quadratic Forms. . . . 30

1.4 Optimal Embeddings . . . 32

2 Elliptic Curves **37**

2.1 Elliptic Curves and Isogenies . . . 37

2.2 The Endomorphism Ring . . . 43

2.2.1 The Trace of Frobenius . . . 47

2.2.2 Action by the Class Group . . . 49

2.3 Supersingularity . . . 51

2.3.1 Properties of Supersingular Curves . . . 52

2.3.2 Orientations and Action by the Class Group . . . 54

2.4 Level Structures . . . 56

3 The Deuring Correspondence 61

3.1 Just Enough Category Theory . . . 61

3.2 An Equivalence of Categories . . . 63

3.3 Computing the Deuring Correspondence . . . 67

3.3.1 The Efficient Direction and the KLPT Algorithm . . . 68

3.3.2 The Hard Direction . . . 70

3.4 Optimal Embeddings and Orientations . . . 72

3.4.1 Quaternion Ideals Generated by Quadratic Ideals . . . 72

3.4.2 Class Group Action on the Oriented Class Set . . . 73

4 Cryptography from Endomorphism Rings 77

4.1 Cryptographic Group Actions . . . 77

4.1.1 CRS . . . 81

4.1.2 CSIDH . . . 81

4.1.3 SCALLOP . . . 83

4.2 SQIsign . . . 83

4.3 Other Isogeny-Based Protocols . . . 85

II The Constructive Deuring Correspondence 97

Paper 1: Deuring for the People: Supersingular Elliptic Curves with Prescribed Endomorphism Ring in General Characteristic 99

Paper 2: Cryptographic Smooth Neighbors 145

Paper 3: AprèsSQI: Extra Fast Verification for SQIsign Using Extension-Field Signing 185

III Oriented Endomorphism Rings 231

Paper 4: Computing Orientations from the Endomorphism Ring of Supersingular Curves and Applications 233

Paper 5: Generalized Class Group Actions on Oriented Elliptic Curves with Level Structure 267

Paper 6: PEARL-SCALLOP: Parameter Extension Applicable in Real Life for SCALLOP 291

Introduction

One of the most powerful principles in mathematics is that of duality, where two seemingly unrelated worlds turn out to essentially be the same, at least structurally. This allows us to translate problems we have no idea how to tackle in one world to a corresponding problem we know how to solve in another world. There are countless examples of such dualities, but the one that lies at the centre of this thesis is the bridge which connects the world of supersingular elliptic curves to the world of quaternion algebras. This bridge is referred to as the Deuring correspondence, after Max Deuring, who made the first connection between these worlds [36]. This seemingly abstract correspondence has recently found a very concrete practical application, in the context of cryptography.

Elliptic curves already have a long history of proving useful in the context of cryptography. For instance, whenever you visit a web page today, your computer is likely executing a cryptographic protocol which involves computing in the group of rational points on an elliptic curve.[3] However, these protocols, like virtually all (asymetric) cryptographic protocols used today, always rely on the hardness of some variant of the discrete logarithm problem.[4] This is despite the fact that it has been known since 1994 that this problem is computationally easy when given access to a quantum computer [70].

As of today, no quantum computer big enough to break these cryptosystems is known to exist, as building such a quantum computer seems to be a huge challenge in practice. However, recent advancements have made the *threat* of a quantum computer existing in the near future big enough that most of the world is currently preparing for a shift into using post-quantum cryptography; cryptography based on hard problems which are presumably hard, even for quantum computers.

One particular flavour of post-quantum cryptography is isogeny-based cryptography, which uses other problems from the theory of elliptic curves to build cryptosystems. Specifically, since cryptography that “happens” in the group of rational points on a fixed elliptic curve is broken by quantum computers, a next step is to consider maps between

[3] For instance, a scan from 2017 estimated that over 70 percent of TLS hosts at least *support* elliptic curve cryptography [76, Table 1], and the number is likely even higher today.

[4] At least if we include finding the order of a group element as a type of discrete logarithm problem, to also cover factorization.

elliptic curves. Such maps are called isogenies, and more or less all (modern) variants of isogeny-based cryptosystems rely on a version of the following problem:

Given two supersingular elliptic curves, find an isogeny between them.

Here, the importance of the Deuring correspondence becomes immediate: While this problem seems very hard, the corresponding problem under the Deuring correspondence is the problem below, which turns out to be easy.

Given two maximal orders, find a connecting ideal.

Luckily, the Deuring correspondence seems to be, at least computationally, a one-way bridge. With the help of the KLPT algorithm [48], going from the world of quaternion algebras to elliptic curves can be done efficiently. However, going in the other direction is, seemingly, a hard problem, which all of isogeny-based cryptography relies on. Historically, the first practical application of the KLPT algorithm in cryptography was to *break* a hash-function based on isogenies [10], when not instantiated carefully. However, modern protocols also use it constructively, i.e. to *make* cryptography.

Outline and Summary of Contributions

This thesis consists of three parts, where Part II and Part III each consist of three papers.

Part I. In the first part of this thesis, we go over the mathematical background required for understanding Part II and Part III.

Chapter 1 focuses on the integral theory of two very related central simple $\mathbb{Q}$-algebras, namely imaginary quadratic fields and definite quaternion algebras, thus introducing the first side of the Deuring correspondence. The chapter also combines these two topics by briefly covering some parts of the theory of optimal embeddings.

Chapter 2 focuses on the other side of the Deuring correspondence, introducing elliptic curves and their isogenies. In particular, the focus lies on endomorphism rings of elliptic curves over finite fields, and naturally, supersingularity. This also includes an overview of the well-known class group action on a set of elliptic curves with complex multiplication (CM-curves), as well as on primitively oriented elliptic curves.

Chapter 3 focuses fully on the Deuring correspondence and connects the material in Chapter 1 and Chapter 2. We also discuss the quaternion analogue of the class group action from the previous chapter, and show that the Deuring correspondence thus can give an alternative proof of this class group action.

Chapter 4 is concerned with applications of the material in the previous chapters, with a focus on the parts of isogeny-based cryptography that are most relevant to the rest of this thesis, as well as a short overview of recent developments that go beyond the scope of this thesis.

Part II. The second part of the thesis consists of three papers, all with a focus on computing the Deuring correspondence in the computationally efficient direction.

The key to making the Deuring correspondence efficient is the KLPT algorithm [48], which is a purely quaternionic algorithm for finding elements of a given norm in quaternion lattices (see also Section 3.3.1 in Part I, for a description of how computing the Deuring correspondence relates to KLPT). Already in the original paper [48, Section 5], it was noted that

> *...we expect our results to lead to a constructive version of Deuring's correspondence ...*

However, in the most general case, attempts at algorithms to compute this showed it to be a significant challenge in practice [45, 62].

> **Paper 1 – Deuring for the People: Supersingular Elliptic Curves with Prescribed Endomorphism Ring in General Characteristic.** *Jonathan Komada Eriksen, Lorenz Panny, Jana Sotáková, and Mattia Veroni.* In this paper, we consider the problem of making the Deuring correspondence constructive: given a maximal order in a quaternion algebra ramified at p and ∞, compute the corresponding elliptic curve. Previous implementations for computing the Deuring correspondence either required p to be very carefully chosen, or they were unable to handle sizes of p more than a few tens of bits. Our algorithm follows the same overall strategy, but by including several optimisations, including a longer discussion of computing rational isogenies from irrational generator points, we provide the first implementation that can solve this problem, even for cryptographically sized primes p.

In many applications of the Deuring correspondence, one works over a fixed characteristic p, where p is allowed to be chosen beforehand. The choice of this prime p heavily influences the efficiency of the computation of the Deuring correspondence. One example of such an application is the signature scheme SQIsign [30, 32]. In SQIsign, one requires that all computation happens over $\mathbb{F}_{p^2}$, which leads to a situation where the efficiency of the scheme essentially comes down to choosing a prime p such that $p^2 - 1$ has a large, *smooth* divisor T, where a smooth number loosely refers to a number which is only divisible by small primes. A related, more classical problem, is that of finding a twin-smooth integer m, i.e. an integer m such that both m and $m + 1$ are smooth. This problem first appeared in isogeny-based cryptography in the context of finding parameters to the (now broken) key-exchange protocol B-SIDH [22, 23]. An older algorithm, which produces smooth numbers with better smoothness bounds is the one by Conrey, Holmstrom, and McLaughlin [20]. However, it is infeasible to compute cryptographically sized twin-smooths with this algorithm.

> **Paper 2 – Cryptographic Smooth Neighbors.** *Giacomo Bruno, Maria Corte-Real Santos, Craig Costello, Jonathan Komada Eriksen, Michael Meyer, Michael Naehrig, and Bruno Sterner.*
> In this paper, we revisit the algorithm by Conrey, Holmstrom, and McLaughlin [20] for finding twin-smooth numbers. We show how the algorithm is simply based on multiplication in 2-dimensional $\mathbb{Q}$-algebras, and thus generalises to finding smooth values of arbitrary quadratic polynomials. We further provide an optimised implementation to compute record-sized twin-smooth numbers. Even though these twin-smooths are still not big enough to be used directly in cryptography, we show how to "boost" such twin-smooth numbers to create primes that are suitable for use in SQIsign, particularly those geared towards higher security levels.

As a signature scheme, SQIsign has remarkably slow signing times. On the other hand, verification is relatively fast (e.g. the second version reports verification times around 6 ms [32]), and the combined size of the public key and signature is by far the best among all known post-quantum signatures [12]. This makes SQIsign especially suited for long-term signatures, where a single signature is expected to be verified many times, and hence optimising verification is espacially important for SQIsign.

Despite the efforts in Paper 2, finding great parameters for SQIsign is very hard, especially those that lead to faster verification, as skewing the parameters towards fast verification quickly makes signing infeasibly slow. This is, in large part, due to the strict requirement of staying over $\mathbb{F}_{p^2}$. However, by applying some of the optimisations from Paper 1, computing isogenies from kernels generated by points in bigger extension fields might be less costly than first assumed.

> **Paper 3 – AprèsSQI: Extra Fast Verification for SQIsign Using Extension-Field Signing.** *Maria Corte-Real Santos, Jonathan Komada Eriksen, Michael Meyer, and Krijn Reijnders.*
> In this paper, we optimise the verification procedure in SQIsign. By allowing for extension field arithmetic in the signature procedure, we get a much looser requirement on the underlying characteristic. While this does not a priori give faster signing in SQIsign, the increased flexibility allows for tweaking the parameters to make verification much faster. We show that this allows for verification that is more than 2 times faster, or almost 4 times faster with size-speed trade-offs, without making signing significantly slower.

Part III. In the third and final part, we consider optimal embeddings, which are embeddings of quadratic orders into quaternion orders, and primitive orientations, which are optimal embeddings into the endomorphism ring of a (necessarily supersingular) elliptic curve. Orientations, among other things, give more structure to isogeny graphs.

Paper 4 – Computing Orientations from the Endomorphism Ring of Supersingular Curves and Applications. *Jonathan Komada Eriksen and Antonin Leroux.*
In this paper, we consider the problem of computing (optimal) embeddings. This problem is equivalent to finding integer representations of ternary quadratic forms, a classical problem. We develop new algorithms for solving this, asymptotically increasing the range for which this problem is solvable in polynomial time. Furthermore, we show how this problem, perhaps surprisingly, relates to problems that occur naturally in the context of isogeny-based cryptography, even problems which at first glance have nothing to do with orientations. As an example, we show how to use our algorithms for solving the embedding problem to find optimal paths in isogeny graphs between very special curves and increasing the range for which this is efficient for more general cases.

Orientations also generalise parts of the classical CM theory of ordinary elliptic curves to supersingular curves. This theory gives a strong relationship between the class groups of imaginary quadratic orders, and certain isomorphism classes of elliptic curves. Both sides of this relationship can be made more fine-grained: On the one hand, one can construct *generalised* class groups, which are more general quotients of the so-called ray-class group, while on the other hand, the elliptic curves may be equipped with a level structure (see Section 2.4 in Part I). The latter topic, in particular, has recently gained more attention in isogeny-based cryptography (see Section 4.3 in Part I for an overview).

Paper 5 – Generalized Class Group Actions on Oriented Elliptic Curves with Level Structure. *Sarah Arpin, Wouter Castryck, Jonathan Komada Eriksen, Gioella Lorenzon and Fréderik Vercauteren.*
In this paper, we generalise the well-known class group action on oriented elliptic curves, and ordinary elliptic curves with complex multiplication. Specifically, we show how a large class of *generalised* class groups act on the same curves, with the additional information of a suitable level structure. As a special example of our result, we interpret the usual action of a non-maximal order on the set of $\mathfrak{O}$-oriented curves to be equivalent to acting on the set of $\mathfrak{O}_K$-oriented curves together with cyclic subgroups generating isogenies whose codomains are $\mathfrak{O}$-oriented.

Finally, primitive orientations have a very concrete cryptographic application in SCALLOP [31], an effective cryptographic group action, (see Section 4.1 in Part I), which is arguably the most natural post-quantum generalisation of the classical Diffie–Hellman primitive. Still, in practice, the performance of SCALLOP is unacceptably slow: In the original work, a single group action evaluation took over 12 minutes for security level equivalent to CSIDH-1024 (a security level which arguably does not offer sufficient resistance against quantum computers [11]), and instantiations of higher security levels were out of reach [31, Section 6].

> **Paper 6 – PEARL-SCALLOP: Parameter Extension Applicable in Real Life for SCALLOP.** *Bill Allombert, Márton Tot Bagi, Jean-françois Biasse, Jonathan Komada Eriksen, Péter Kutas, Chris Leonardi, Aurel Page and Renate Scheidler.*
> In this paper, we consider a novel way to instantiate SCALLOP, by relaxing two of the requirements on the quadratic order: We allow the maximal order on top to have a non-trivial class group, and we allow the conductor to be a product of large primes, instead of a prime. We show how these relaxed requirements give us sufficient freedom to find parameters that are ideal for SCALLOP, while still being able to compute the class group structure. Compared to SCALLOP, our version is easier to instantiate for higher security levels, at least an order of magnitude faster, and based on a slightly different security assumption. While the new security assumption may introduce new avenues of attack, we also argue that it just as well avoids potential attacks on the original SCALLOP security assumption.

Part 1

Background

He climbed on top of a giant's limb
to see further, despite being quite dim.
He opened his eye,
and started to cry,
'cause the fog was thicker than him!

Jonathan Komada Eriksen,
April 2024

Chapter 1

Central Simple Algebras over the Rationals

We start by studying two classes of central division algebras over $\mathbb{Q}$, namely imaginary quadratic number fields, and (definite) quaternion algebras over $\mathbb{Q}$. Inside these algebras, we are especially interested in orders and their ideals, because of their relation to elliptic curves and isogenies.

1.1 Preliminary Definitions

First, we cover a few fundamental definitions and results, which will apply in both cases of interest. While general results and definitions may at times be harder to apply than more "concrete" results, we gain valuable insight into *why* the two cases we are interested in are often analogous.

Definition 1.1.1. Let F be a field, and let A be a ring. We say that A is an ***F*-algebra**, if F embeds into $Z(A)$, the center of A. Further

(i) A is **central** if A is finite-dimensional over F, and $F \cong Z(A)$

(ii) A is **simple** if A is simple as a ring.

(iii) A is a **division algebra** if every non-zero element of A has an inverse.

Clearly, if A is a division algebra over F, it is also a simple F-algebra. While these are not the only simple F-algebras, Wedderburn's theorem famously says that an F-algebra A is central and simple if and only if it is isomorphic to a matrix algebra $\mathbf{M}_n(K)$, where K is a central division algebra over F.

In this thesis, we will be interested in two cases of central simple $\mathbb{Q}$-algebras, namely:

- **Number Fields**, which are central division algebras over $\mathbb{Q}$.
- **Quaternion Algebras** over $\mathbb{Q}$, which are either central division algebras over $\mathbb{Q}$, or isomorphic to $\mathbf{M}_2(\mathbb{Q})$.

Restricting to central division F-algebras, our cases of interest are characterised by the existence of a standard involution, which we define below

Definition 1.1.2. Let A be an F-algebra. An **involution** is an F-linear map

$$\bar{}\colon A \to A,$$

satisfying the following

(i) $\overline{1} = 1$,

(ii) $\overline{\overline{\alpha}} = \alpha$ for all $\alpha \in A$,

(iii) $\overline{(\alpha\beta)} = (\overline{\beta})(\overline{\alpha})$.

Further, an involution is **standard** if

(iv) $\alpha\overline{\alpha} \in F$ for all $\alpha \in A$.

One can show that a standard involution also satisfies $\alpha + \overline{\alpha} \in F$, where $\alpha\overline{\alpha} \in F$ for all $\alpha \in A$. With this in mind, we define the reduced norm $\operatorname{nrd}(\alpha)$ and reduced trace $\operatorname{trd}(\alpha)$ for $\alpha \in A$ to be

$$\begin{aligned} \operatorname{nrd}(\alpha) &:= \alpha\overline{\alpha}, \\ \operatorname{trd}(\alpha) &:= \alpha + \overline{\alpha}. \end{aligned}$$

When A possesses a standard involution, every element in A is the root of a quadratic polynomial in $F[X]$. Specifically, $\alpha \in A$ is a root of

$$X^2 - \operatorname{trd}(\alpha)X + \operatorname{nrd}(\alpha).$$

Famously, the only finite-dimensional division $\mathbb{R}$-algebras are $\mathbb{R}$, $\mathbb{C}$ and $\mathbb{H}$, where $\mathbb{H}$ denotes Hamilton's "original" quaternions. When restricting to division $\mathbb{Q}$-algebras with a standard involution, we get the analogous result we state below.[1] Note that the theorem below is not only interesting in its own right; it is also the key to proving Proposition 2.2.4, the first step in connecting Chapter 1 and Chapter 2.

Theorem 1.1.3. *Let A be a division $\mathbb{Q}$-algebra, with a standard involution. Then one of the following holds.*

[1] In fact, the theorem is, with slight adjustments, true for general F-algebras, hence truly generalising the case of $F = \mathbb{R}$

(i) $A = \mathbb{Q}$.

(ii) *A is a quadratic number field.*

(iii) *A is a qaternion algebra over $\mathbb{Q}$ (not isomorphic to $\mathbf{M}_2(\mathbb{Q})$).*

Proof. See [78, Theorem 3.5.1]. □

In the case (ii) of Theorem 1.1.3, the reduced norm and trace coincides with the usual norm and trace in $\mathbb{Q}$-algebras, hence we will typically denote $\mathrm{nrd}(\alpha)$ as $\mathrm{n}(\alpha)$ and $\mathrm{trd}(\alpha)$ as $\mathrm{t}(\alpha)$ in those cases. However, in case (iii), these are not the same; in fact, one can show that $\mathrm{nrd}(\alpha)^2 = \mathrm{n}(\alpha)$ and $2\mathrm{trd}(\alpha) = \mathrm{t}(\alpha)$ for all $\alpha \in A$ in that case.

One fundamental theorem of central simple F-algebras, which we will apply many times, is the following:

Theorem 1.1.4 (Skolem–Noether)**.** *Let A and B be central simple F-algebras. Given F-algebra homomorphisms*

$$f, g : A \to B,$$

there exists an element $\beta \in B^\times$ such that

$$g(\alpha) = \beta^{-1} f(\alpha) \beta,$$

for all $\alpha \in A$.

Proof. See [78, Main Theorem 7.7.1]. □

Typically, for our case of interest, we will work with a fixed algebra A, and look at integral objects sitting inside of A. This theory looks similar in both cases, but with important differences which we will highlight in the two next sections. The generic definitions are the following, though keep in mind that we are always in the case $n = 2$ or $n = 4$.

Definition 1.1.5. Let A be a central simple $\mathbb{Q}$-algebra with $\dim A = n < \infty$.

- A **lattice** L in A is a finitely generated $\mathbb{Z}$-submodule $L \subseteq B$ such that $L \otimes \mathbb{Q} = A$. Equivalently, it is a $\mathbb{Z}$-submodule $L \subseteq B$ of rank n.
- A lattice $O \subset A$ is an **order** if it is also a subring of A, i.e. if it contains 1 and is closed under multiplication.
- Let $O \subset A$ be an order. A lattice I is a **left (resp. right) fractional O-ideal** if $OI \subseteq I$ (resp. $IO \subseteq I$), implying that I is a left (resp. right) O-module. If $I \subset O$, we simply refer to I as a **left (resp. right) O-ideal**. If I is both a left and right (fractional) O-ideal, e.g. whenever A is commutative, we often drop the "left" and "right".

As the following example shows, these concepts generalise objects which might be more familiar.

Example 1: Lattices, orders and ideals. Consider $A = \mathbb{Q}$ itself as a (rather trivial) central simple $\mathbb{Q}$-algebra. A lattice L in $\mathbb{Q}$ is simply a $\mathbb{Z}$-module in $\mathbb{Q}$ of rank 1, i.e. it looks like something of the form

$$\frac{n}{m}\mathbb{Z} = \left\{ \frac{nk}{m} \mid k \in \mathbb{Z} \right\},$$

for some $\frac{n}{m} \in \mathbb{Q}$. If L is an order, it must contain 1, and be closed under multiplication, hence it is easy to see that the only order in $\mathbb{Q}$ is $\mathbb{Z}$ itself.

Any such lattice L is a fractional $\mathbb{Z}$-ideal, and the (integral) $\mathbb{Z}$-ideals of $\mathbb{Q}$ are the fractional ideals of the form $n\mathbb{Z}$, i.e. the usual ideals of $\mathbb{Z}$. Note that even though we refer to these as "ideals in $\mathbb{Q}$", they are clearly not $\mathbb{Q}$-ideals in the usual sense.[a]

[a] Since $\mathbb{Q}$ is a field, the only $\mathbb{Q}$-ideals are (0) and $\mathbb{Q}$ itself, neither of which satisfy the axioms of a lattice, since (0) does not contain a basis of $\mathbb{Q}$, while $\mathbb{Q}$ itself is not finitely generated as a $\mathbb{Z}$-module.

Next, we look at the two cases of division algebras over $\mathbb{Q}$ with a standard involution we are interested in, separately.

1.2 Imaginary Quadratic Number Fields

A **number field** is an algebraic extension of $\mathbb{Q}$. Our study of number fields K will be limited to **imaginary quadratic** number fields, i.e. number fields of extension degree 2 over $\mathbb{Q}$, which do not embed into $\mathbb{R}$. Even if this sounds like a very specific case, the theory of imaginary quadratic number fields is very rich, as it has been studied a great deal. Most of the material in this section can be found in the book by Cox [26].

Any imaginary quadratic number field K is isomorphic to $\mathbb{Q}(\sqrt{-d})$ for some square-free integer $d > 0$. The **discriminant** of K, which we will denote by d_K, is closely related to this d, with $d_K = -d$ when $d \equiv 3 \pmod 4$, and $d_K = -4d$ otherwise.

For the remainder of this section, let K denote an imaginary quadratic number field. Some statements in this section are true in greater generality, but for ease of exposition, we consider only the case we are interested in.

1.2.1 Imaginary Quadratic Orders and Their Ideals

Recall that an order $\mathfrak{O} \subseteq K$ is a subring of K, that has rank 2 as a $\mathbb{Z}$-module. The fact that the ring $\mathfrak{O} \supset \mathbb{Z}$ is finitely generated as a $\mathbb{Z}$-module is equivalent to the fact that all elements of $\mathfrak{O}$ are integral, i.e. they satisfy a monic polynomial in $\mathbb{Z}[X]$. The integral closure of $\mathbb{Z}$ in K (i.e. the subring of K consisting of all integral elements) is called the **ring of integers** in K, and denoted by $\mathfrak{O}_K$. One can show that $\mathfrak{O}_K$ is an order, and further that all other orders $\mathfrak{O} \subset K$ satisfy $\mathfrak{O} \subseteq \mathfrak{O}_K$.

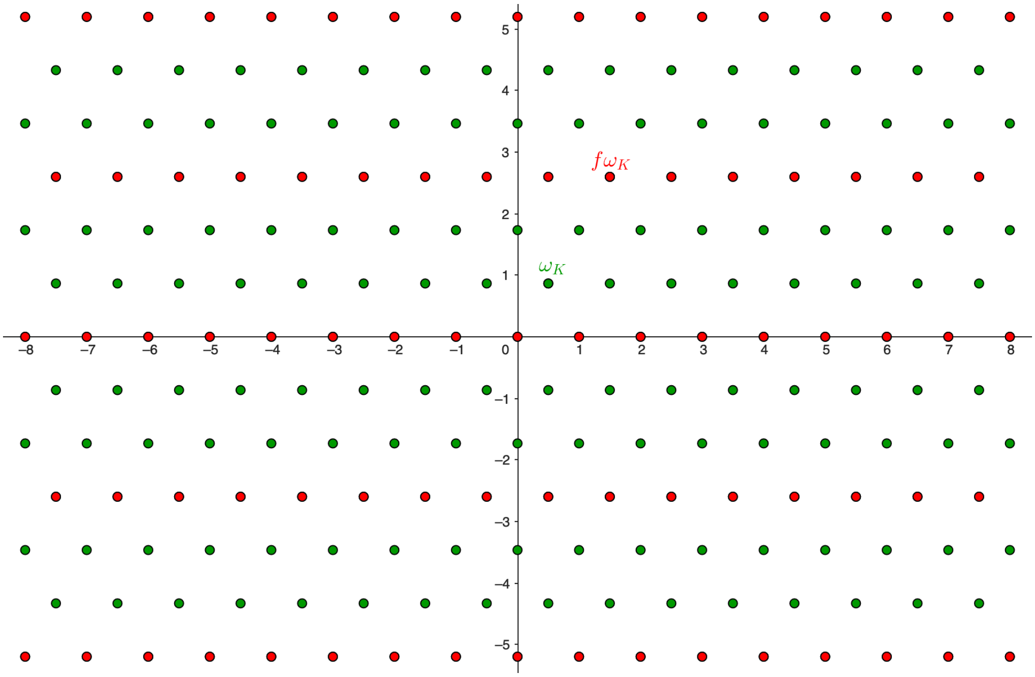


Figure 1.1: The ring of integers $\mathfrak{O}_K = \mathbb{Z}\left[\frac{1+\sqrt{-3}}{2}\right]$ in $K = \mathbb{Q}(\sqrt{-3})$, and the suborder of conductor $f = 3$ in red.

Definition 1.2.1. Let $\mathfrak{O} \subseteq K$ be an order, and let $f = [\mathfrak{O}_K : \mathfrak{O}]$. Then f is called the **conductor** of $\mathfrak{O}$.

The conductor of an order in K uniquely determines the order. The ring of integers $\mathfrak{O}_K$ can always be written as $\mathbb{Z}[\omega_K]$, where

$$\omega_K = \begin{cases} \sqrt{\frac{d_K}{2}} & d_K \equiv 0 \pmod 4, \\ \frac{1+\sqrt{d_K}}{2} & d_K \equiv 1 \pmod 4, \end{cases}$$

which implies that $\mathfrak{O} = \mathbb{Z} + f\mathfrak{O}_K = \mathbb{Z}[f\omega_K]$. See Figure 1.1 for an illustration in $\mathbb{Q}(\sqrt{-3})$.

Letting $\mathfrak{O} \subseteq K$ be an order of conductor f, we can define the **discriminant** of $\mathfrak{O}$ to be the value $f^2 d_K$.

Fractional Ideals. Every $\mathbb{Z}$-lattice L in K is of the form

$$\alpha\,\mathbb{Z} + \beta\,\mathbb{Z}, \quad \alpha, \beta \in K$$

and, by definition, such a lattice can be considered a fractional $\mathfrak{O}$-ideal for any

$$\mathfrak{O} \subseteq \{\gamma \in K \mid \gamma L \subseteq L\}.$$

Note that we need not have equality here, which motivates the following definition.

Definition 1.2.2. Let $\mathfrak{a} \subset K$ be a fractional $\mathfrak{O}$-ideal. Then $\mathfrak{a}$ is said to be **proper** if

$$\mathfrak{O} = \{\gamma \in K \mid \gamma \mathfrak{a} \subseteq \mathfrak{a}\}.$$

The notion of being a proper fractional ideal extends naturally to fractional ideals in $\mathbb{Q}$-algebras, see Definition 1.3.9 for the case of quaternion algebras.

In fact, one can show that $\{\gamma \in K \mid \gamma L \subseteq L\}$ is an order for any $\mathbb{Z}$-lattice L. Thus it is immediately obvious that every $\mathfrak{O}_K$-ideal is proper. We will mainly be dealing with proper ideals, though there is one notable exception that arises in Remark 2.

We extend the notion of **norm** to (proper) fractional ideals, by defining

$$\mathrm{n}(\mathfrak{a}) = \gcd(\{\mathrm{n}(\alpha) \mid \alpha \in \mathfrak{a}\}).$$

The norm extends the notion of norm of elements, i.e. $\mathrm{n}(\alpha\mathfrak{O}) = \mathrm{n}(\alpha)$, and is multiplicative, i.e. $\mathrm{n}(\mathfrak{a}\mathfrak{b}) = \mathrm{n}(\mathfrak{a})\mathrm{n}(\mathfrak{b})$ [26, Lemma 7.14].

Prime splitting. For the rest of this subsection, we look at the ring of integers $\mathfrak{O}_K$, and specifically, the set of prime ideals in $\mathfrak{O}_K$, denoted by $\operatorname{Spec}\mathfrak{O}_K$. Rings of integers $\mathfrak{O}_K$ are Dedekind domains, which are characterised by having unique factorisation of ideals, i.e. every $\mathfrak{O}_K$-ideal $\mathfrak{a}$ can be written as

$$\mathfrak{a} = \mathfrak{p}_1^{e_1}\mathfrak{p}_2^{e_2}\dots\mathfrak{p}_n^{e_n}, \quad \mathfrak{p}_i \in \operatorname{Spec}\mathfrak{O}_K, e_i \in \mathbb{Z}_{>0},$$

in a unique way (up to ordering). Note that this fails for orders $\mathfrak{O} \subsetneq \mathfrak{O}_K$, as these are not integrally closed (hence, by definition, not Dedekind domains).

Given a prime $p \in \mathbb{Z}$, the ideal $p\mathbb{Z}$ is, of course, a prime ideal in $\mathbb{Z}$, but lifting this ideal to $\mathfrak{O}_K$, i.e. considering the ideal $p\mathfrak{O}_K$, it need not remain prime in $\mathfrak{O}_K$. We have three cases to consider.

Proposition 1.2.3. *Let $p \in \mathbb{Z}$ be a prime, and let d_K denote the discriminant of K. Then*

$$p\mathfrak{O}_K = \begin{cases} \mathfrak{p}^2, \textit{for some } \mathfrak{p} \in \operatorname{Spec}\mathfrak{O}_K, & \Leftrightarrow \left(\frac{d_k}{p}\right) = 0 \\ \mathfrak{p}\mathfrak{p}', \textit{for some } \mathfrak{p}, \mathfrak{p}' \in \operatorname{Spec}\mathfrak{O}_K, & \Leftrightarrow \left(\frac{d_k}{p}\right) = 1 \\ p\mathfrak{O}_K, \textit{and } p\mathfrak{O}_K \in \operatorname{Spec}\mathfrak{O}_K, & \Leftrightarrow \left(\frac{d_k}{p}\right) = -1 \end{cases}$$

Proof. [26, Proposition 5.16]. □

For the three cases above, we call the prime p **ramified**, **split** and **inert**, respectively. In the second case, the prime ideals $\mathfrak{p}, \mathfrak{p}'$ are Galois conjugates. Note that since K has degree 2 over $\mathbb{Q}$, Proposition 1.2.3 is a simpler case of the more general theorem of prime splitting in Galois extensions, and primes are either completely ramified, completely split or inert.

> **Example 2: Primes splitting.** Let $K = \mathbb{Q}(\sqrt{-23})$, such that $\mathfrak{O}_K = \mathbb{Z}[\delta], \delta = \frac{1+\sqrt{-23}}{2}$. Considering the ideal
>
> $$5\mathfrak{O}_K = 5\mathbb{Z} + \frac{5+5\sqrt{-23}}{2}\mathbb{Z},$$
>
> we know from Proposition 1.2.3 that $5\mathfrak{O}_K \in \operatorname{Spec}\mathfrak{O}_K$, since $\left(\frac{-23}{5}\right) = -1$. However, the ideal $3\mathfrak{O}_K$ should split, as $\left(\frac{-23}{3}\right) = 1$. And sure enough, we find that $3\mathfrak{O}_K = \mathfrak{p}\mathfrak{p}'$, where
>
> $$\begin{aligned} \mathfrak{p} &= 3\mathbb{Z} + \frac{1+\sqrt{-23}}{2}\mathbb{Z}, \\ \mathfrak{p}' &= 3\mathbb{Z} + \frac{1-\sqrt{-23}}{2}\mathbb{Z}. \end{aligned}$$

1.2.2 The Class Group

Next, we will give the set of invertible fractional ideals of an order a group structure, which will be of huge importance in this thesis. A fractional $\mathfrak{O}$-ideal $\mathfrak{a}$ is said to be **invertible** if there exists some other fractional ideal $\mathfrak{a}^{-1}$, such that $\mathfrak{a}\mathfrak{a}^{-1} = \mathfrak{O}$. The invertible ideals coincide with one of our earlier definitions.

Proposition 1.2.4. *Let $\mathfrak{a}$ be a fractional $\mathfrak{O}_K$-ideal. Then $\mathfrak{a}$ is invertible if and only if $\mathfrak{a}$ is proper.*

Proof. See [26, Proposition 7.4]. □

If we now restrict to the ring of integers again, we immediately see that all fractional $\mathfrak{O}_K$-ideals are invertible (since all fractional $\mathfrak{O}_K$-ideals are proper). One can then show that any fractional $\mathfrak{O}_K$-ideal has a unique factorization of the form

$$\mathfrak{a} = \mathfrak{p}_1^{r_1}\mathfrak{p}_2^{r_2}\dots\mathfrak{p}_n^{r_n}, \quad \mathfrak{p}_i \in \operatorname{Spec}\mathfrak{O}_K, r_i \in \mathbb{Z},$$

up to ordering of course. From this, it follows that the set of fractional $\mathfrak{O}_K$-ideals, denoted by $I(\mathfrak{O}_K)$, is closed under multiplication, and by Proposition 1.2.4, it is also closed under inverses, hence it is a group. Further, again by unique factorization of ideals, it is clear that the set $P(\mathfrak{O}_K) \subseteq I(\mathfrak{O}_K)$ of **principal** fractional $\mathfrak{O}_K$-ideals is a subgroup, allowing us to make the following definition.

Definition 1.2.5. The **class group** of $\mathfrak{O}_K$, denoted $\mathrm{Cl}(\mathfrak{O}_K)$ is the quotient group

$$\mathrm{Cl}(\mathfrak{O}_K) := I(\mathfrak{O}_K)/P(\mathfrak{O}_K).$$

Note that clearly $\mathrm{Cl}(\mathfrak{O}_K)$ is trivial if and only if $\mathfrak{O}_K$ is a principal ideal domain. Interestingly, it turns out that for imaginary, quadratic fields, $\mathrm{Cl}(\mathfrak{O}_K)$ is trivial if and only if

$$d_K \in \{-3, -4, -7, -8, -11, -19, -43, -67, -163\},$$

see [26, Theorem 7.30 (i)].

To extend the notion of class group to generic orders $\mathfrak{O}$, we need the following proposition, which states that the only place where stuff goes wrong is at primes dividing the conductor.

Proposition 1.2.6. *Let $\mathfrak{O}$ be an order of conductor f, and let $\mathfrak{a}$ be a fractional $\mathfrak{O}$-ideal with $\gcd(\mathrm{n}(\mathfrak{a}), f) = 1$. Then $\mathfrak{a}$ is invertible.*

Proof. Combine [26, Lemma 7.18 (ii)] and [26, Proposition 7.4]. □

Hence, we can consider the set $I(\mathfrak{O}, f)$ of fractional $\mathfrak{O}$-ideals of norm prime to f; it is clear that this is again a group.[2] If we further consider the subgroup of principal fractional $\mathfrak{O}$-ideals prime to f, denoted $P(\mathfrak{O}, f)$, we can extend the definition of class groups to arbitrary orders by defining

$$\mathrm{Cl}(\mathfrak{O}) := I(\mathfrak{O}, f)/P(\mathfrak{O}, f).$$

Remark 1. *More generally, we can form the class group of any order $\mathfrak{O}$ as the quotient of the invertible ideals modulo the principal ones. This construction is isomorphic to the one we made above. However, in general, working with ideals prime to the conductor is easier, for instance, because one can show that these have unique factorization, so we take this as our definition. This will cause no trouble, because of a result that says that given any fractional $\mathfrak{O}$-ideal $\mathfrak{a}$, and integer m, we can find another ideal $\mathfrak{b}$ prime to m, with* $[\mathfrak{b}] = [\mathfrak{a}]$ *[26, Corollary 7.17].*

There is an obvious map from $I(\mathfrak{O}, f) \to I(\mathfrak{O}_K)$, sending the fractional $\mathfrak{O}$-ideal $\mathfrak{a}$ to $\mathfrak{a}\mathfrak{O}_K$. What is much less obvious (but nevertheless, true) is that this map induces a surjection on the level of class groups and that we can accurately describe the kernel, resulting in the following exact sequence.

Theorem 1.2.7. *Let $\mathfrak{O} \subset \mathfrak{O}_K$ be an order of conductor f. Then we have the following exact sequence:*

$$1 \to \mathfrak{O}_K^\times/\mathfrak{O}^\times \to (\mathfrak{O}_K/f\mathfrak{O}_K)^\times/(\mathfrak{O}/f\mathfrak{O}_K)^\times \to \mathrm{Cl}(\mathfrak{O}) \to \mathrm{Cl}(\mathfrak{O}_K) \to 1$$

[2] In general, $I(\mathfrak{O}, N)$ refers to the group of invertible fractional $\mathfrak{O}$-ideals coprime to N, but when $N = f$ is the conductor, the “invertible” is reduntant in light of Proposition 1.2.6

Proof. The sequence can be derived from the material in Cox [26, §7], or alternatively, see proposition 3.10 in Paper 5 for a proof of a more general statement. □

Remark 2. *Note that* $(\mathfrak{O}/f\mathfrak{O}_K)^\times$ *is in fact not a typographical error! Clearly,* $f\mathfrak{O}_K$ *is an* $\mathfrak{O}$*-ideal, though, notably, not a proper (Definition 1.2.2)* $\mathfrak{O}$*-ideal, hence the quotient* $(\mathfrak{O}/f\mathfrak{O}_K)$ *makes sense.*

1.2.3 Binary Quadratic Forms

We now recall the correspondence between K-lattices and positive definite binary quadratic forms. There are two main reasons that we do this. The first is that we will, towards the end of this section, state Dirichlet's composition laws, and see that this gives a very explicit way of computing in the ideal class group. The second is that it prepares us for Section 1.3.3, which relates quaternion orders and ideals to ternary and quartic quadratic forms.

Definition 1.2.8. An **integral n-ary quadratic form** (over $\mathbb{Z}$) is a homogenous polynomial in n variables with coefficients in $\mathbb{Z}$. Further, the quadratic form is said to be **primitive** if the greatest common divisor of all its coefficients is 1.

In this section, we will stick with binary quadratic forms, i.e. a homogenous polynomial in two variables

$$f(x,y) = ax^2 + bxy + cy^2 \in \mathbb{Z}[x,y].$$

For the rest of this section, let $\mathfrak{O} \subset K$ be an imaginary quadratic order with discriminant $\Delta = f^2 d_K$. Given a lattice $\Lambda = \alpha\mathbb{Z} + \beta\mathbb{Z}$ in K, one can obtain an associated quadratic form

$$f(x,y) = \mathrm{n}(x\alpha + y\beta). \tag{1.1}$$

This associated quadratic form is only well defined up to change of basis of Λ. Corresponding to such a basis change, we define an **equivalence** between two forms $f(x,y), g(x,y)$ to be a matrix $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathbf{GL}_2(\mathbb{Z})$ such that

$$g(x,y) = f(ax+b, cx+d).$$

Further, if $M \in \mathbf{SL}_2(\mathbb{Z})$, it is called a **proper equivalence**.

Theorem 1.2.9. *Let* $f(x,y) = ax^2 + bxy + c$ *be a primitive integral binary quadratic form of discriminant* Δ*, and let* $\mathfrak{a} = \alpha\mathbb{Z} + \beta\mathbb{Z}$ *be an integral* $\mathfrak{O}$*-ideal. The associations sending*

$$f(x,y) \to a\mathbb{Z} + \frac{-b+\sqrt{\Delta}}{2}\mathbb{Z}$$

(where $\sqrt{\Delta}$ is chosen to lie in $\mathfrak{h}$, the upper half plane) and

$$\mathfrak{a} \to g(x,y) = \mathrm{n}(\alpha x - \beta y)/\mathrm{n}(\mathfrak{a})$$

define mutually inverse bijections between the ideal class group $\mathrm{Cl}(\mathfrak{O})$ *and* $\mathrm{Cl}(\Delta)$*, the set of* $\mathbf{SL}_2(\mathbb{Z})$*-equivalence classes of primitive integral binary quadratic forms of discriminant* Δ*.*

Proof. We leave all the details of the proof to [26, Theorem 7.7] and show only that these associations are mutually inverse, as it shows where the form $g(x,y)$ comes from.

To start, let $\tau_f \in \mathfrak{h} \cap K$ be an element satisfying $f(\tau_f, 1) = 0$. Then, we can write the ideal associated to f as $a(\mathbb{Z} + \tau_f\mathbb{Z})$.

Thus, when given an ideal $\mathfrak{a} = \alpha\mathbb{Z} + \beta\mathbb{Z}$, the associated $g(x,y)$ should arise from the minimal polynomial (over $\mathbb{Z}$) of $\tau = \beta/\alpha$. To see this, let $g(x) = ax^2 + bx + c$ be the minimal polynomial of τ. One can show that $g(x)$ has discriminant Δ, and further, the ideal associated to $g(x,y) = ax^2 + bxy + cy^2$ is equivalent to $\mathfrak{a}$, as

$$a(\mathbb{Z} + \tau_g\mathbb{Z}) = a(\mathbb{Z} + \beta/\alpha\mathbb{Z}) = \mathfrak{a}a/\alpha.$$

Finally, it remains to show that $\mathrm{n}(\alpha x - \beta)/\mathrm{n}(\mathfrak{a})$ is indeed the minimal polynomial of β/α. Clearly, $h(x) := \mathrm{n}(\alpha x - \beta)$ is an integer multiple of the minimal polynomial. Explicitly, we can write

$$h(x) = \mathrm{n}(\alpha x - \beta) = \mathrm{n}(\alpha)x^2 - \mathrm{t}(\alpha\bar{\beta})x - \mathrm{n}(\beta),$$

whose gcd is exactly $\mathrm{n}(\mathfrak{a})$, and thus the minimal polynomial of β/α is $h(x)/\mathrm{n}(\mathfrak{a})$. □

The suggestive notation $\mathrm{Cl}(\Delta)$ comes from the fact that we can give this set a natural group structure, such that the association above induces an isomorphism $\mathrm{Cl}(\Delta) \cong \mathrm{Cl}(\mathfrak{O})$. Interestingly, the following group structure was studied long before imaginary quadratic orders and the class group.

Let $f_i(x,y) = a_ix^2 + b_ixy + c_iy^2 \in \mathrm{Cl}(\Delta), i = 1,2$ be two representatives, satisfying $\gcd(a_1, a_2, (b_1+b_2)/2) = 1$ (such representatives can always be found). Then the **composition** of f_1, f_2 is defined to be

$$F(x,y) = a_1a_2x^2 + Bxy + \frac{B^2 - \Delta}{4a_1a_2}y^2 \in \mathrm{Cl}(\Delta),$$

where B satisfies

$$\begin{aligned} B &\equiv b_1 \pmod{2a_1} \\ B &\equiv b_2 \pmod{2a_2} \\ B^2 &\equiv D \pmod{4a_1a_2}. \end{aligned}$$

This composition law turns $\mathrm{Cl}(\Delta)$ into a group, where the identity element is the class containing $x^2 - \frac{\Delta}{4}y^2$ if $\Delta \equiv 0 \pmod 4$, or $x^2 + xy + \frac{1+\Delta}{4}y^2$ if $\Delta \equiv 1 \pmod 4$. Further, one can show that each class has a unique representative $ax^2 + bxy + cy^2$ satisfying

$$(|b| \leq a \leq c) \wedge (((|b| = a) \vee (a = c)) \Rightarrow b \geq 0),$$

i.e. where $|b| \leq a \leq c$, and where b is non-negative if $b = a$ or $a = c$.

Corollary 1.2.10. *The association from Theorem 1.2.9 induces an isomorphism between* $\mathrm{Cl}(\Delta)$ *and* $\mathrm{Cl}(\mathfrak{O})$.

Proof. Again, see [26, Theorem 7.7]. □

From this isomorphism, it is clear where the identity element in the form class group comes from: Writing the discriminant $\Delta = f^2 d_K$, it is simply given by the norm form $g(x, y) = \mathrm{n}(x - f\omega_K y)$ associated to $\mathbb{Z}[f\omega_K]$, where, as before,

$$\omega_K = \begin{cases} \sqrt{\frac{d_K}{2}} & d_K \equiv 0 \pmod 4, \\ \frac{1+\sqrt{d_K}}{2} & d_K \equiv 1 \pmod 4. \end{cases}$$

In the next section, we will use Corollary 1.2.10 to explicitly compute in the ideal class group.

1.2.4 Computing in the Class Group

We recall two natural ways of computing in a given class group through examples. As we will see later, the second one has clear applications in isogeny-based cryptography.

Composition of quadratic forms. One of the simplest ways to compute in the class group of an imaginary quadratic order is precisely by using the composition laws stated towards the end of Section 1.2.3. This is efficient, and does not require knowing the structure of the class group. One application of this is a simple way of computing the class group structure, using generic group-order algorithms [74].[3]

> **Example 3: Composition of binary quadratic forms.** We continue working with $\mathfrak{O}_K = \mathbb{Z}[\omega]$, where $\omega = \frac{1+\sqrt{-23}}{2}$. We start with the ideal $\mathfrak{l} = 3\mathbb{Z} + (\omega - 1)\mathbb{Z}$. Following Theorem 1.2.9, the corresponding quadratic form is
>
> $$g(x, y) = 3x^2 + xy + 2y^2$$
>
> However, notice that this form is not reduced, since $a > c$. To obtain a reduced form,

[3] Note that the generic group-order algorithms are exponential-time in the worst case, while more advanced subexponential algorithms for computing class groups also exist. The first was discovered by Hafner and McCurley [43] and runs in $L(|d_K|)^{\sqrt{2}}$ time on average, see also Cohen [18, Chapter 5].

apply the proper equivalence

$$f(x,y) := \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \cdot g(x,y) = g(-y,x) = 2x^2 - xy + 3y^2.$$

Thus we see that f is not the identity,[a] since it is not of the form $x^2 + xy + \frac{1+\Delta}{4}y^2$.

Next, let us compute $f \star f$ in $\mathrm{Cl}(\Delta)$. We see that $B = 3$ satisfies the congruence conditions from the composition laws, thus

$$f \star f = 4x^2 + 3xy + 2y^2.$$

Again, we can verify that this is not the identity by computing the corresponding reduced form

$$h := \begin{pmatrix} 0 & -1 \\ 1 & 1 \end{pmatrix} \cdot f \star f = 2x^2 + xy + 3y^2.$$

Finally, we compute $f \star f \star f = 8x^2 + 3xy + y^2$. The corresponding reduced form is

$$\begin{pmatrix} 0 & -1 \\ 1 & 2 \end{pmatrix} \cdot f \star f \star f = x^2 + xy + 6y^2,$$

which we see is equal to the identity. Thus, we can conclude that $[\mathfrak{l}] \in \mathrm{Cl}(\mathfrak{O})$ has order 3. This implies that $h(\mathfrak{O}) = \#\,\mathrm{Cl}(\mathfrak{O})$ must divide 3, and further (for instance by using the Minkowski-bound) we can actually conclude that $h(\mathrm{Cl}(\mathfrak{O})) = 3$.

[a] which we knew already, since $\mathfrak{l}$ was not principal.

Working modulo the relation lattice. One very simple way of working in a cyclic group G of known order N is by using the isomorphism from $\mathbb{Z}/N\mathbb{Z}$ given by fixing a generator $g \in G$, and computing

$$\begin{aligned} \iota : \mathbb{Z}/N\mathbb{Z} &\to G \\ \iota(y) &= g^y. \end{aligned}$$

For large N, this is efficient (in this direction) precisely because of the square-and-multiply algorithm [18, Chapter 1.2]. The following can be seen as a generalisation of the idea above.

Let G be a finite, abelian group, and fix any projective resolution

$$0 \to \mathbb{Z}^n \xrightarrow{\gamma} \mathbb{Z}^n \to G \to 0.$$

This gives an isomorphism $G \cong \mathbb{Z}^n/L$, where L is the image of γ (called the lattice of relations in G). Thus, elements of G can be represented by vectors $v \in \mathbb{Z}^n$, and computing the corresponding element in G naïvely takes $||v||_1 - 1$ multiplications. Given any such vector v, we can replace it with an equivalent vector v' of short norm in $\mathbb{Z}^n/L$ by finding the vector e in L closest to v, and setting $v' := e - v$, to minimize the ℓ_1-norm.

Example 4: Working mod the lattice of relations. This time, we work in the order $\mathfrak{O} = \mathbb{Z}[\sqrt{-109}]$. Start with the ideal $\mathfrak{l}_1 = 11\mathbb{Z} + (\sqrt{-109}+1)\mathbb{Z}$. As in Example 3, we use the composition of quadratic forms to find that $\langle[\mathfrak{l}_1]\rangle = \mathrm{Cl}(\mathfrak{O}) \cong \mathbb{Z}/6\mathbb{Z}$. We take two more ideals $\mathfrak{l}_2 = 19\mathbb{Z} + (\sqrt{-109}+9)\mathbb{Z}$ and $\mathfrak{l}_3 = 29\mathbb{Z} + (\sqrt{-109}+6)\mathbb{Z}$. Again, using the form composition, we find the relations

$$[\mathfrak{l}_1]^5 = [\mathfrak{l}_2], \quad [\mathfrak{l}_1]^4 = [\mathfrak{l}_3].$$

Thus, a projective resolution of $\mathrm{Cl}(\mathfrak{O})$ is given by

$$\begin{tikzcd}
0 \arrow[r] & \mathbb{Z}^3 \arrow[r, "\begin{pmatrix} 6 & 0 & 0 \\ 5 & -1 & 0 \\ 4 & 0 & -1 \end{pmatrix}"] & \mathbb{Z}^3 \arrow[r, "\pi"] & \mathrm{Cl}(\mathfrak{O}) \arrow[r] & 0,
\end{tikzcd}$$

where π is determined by sending the basis vector e_i to $[\mathfrak{l}_i]$. By computing a reduced basis of the lattice of relations (for instance using the famous LLL algorithm [52]), we realise our class group as

$$\mathrm{Cl}(\mathfrak{O}) \cong \mathbb{Z}^3/L,$$

where L is the lattice given by the basis

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & -1 & 1 \\ -1 & 1 & 2 \end{pmatrix} \sim \begin{pmatrix} 6 & 0 & 0 \\ 5 & -1 & 0 \\ 4 & 0 & -1 \end{pmatrix}.$$

We now want to compute $[\mathfrak{l}_1]^{12345}$ without square and multiply. Using Babai's nearest plane algorithm [3], we find that the closest vector to $v = [12345, 0, 0]$ in L is $e = [12345, -1, -1]$, and thus we know that $[12345, 0, 0] \equiv [0, 1, 1] \pmod{L}$. Hence, we can simply compute $[\mathfrak{l}_1]^{12345} = [\mathfrak{l}_2][\mathfrak{l}_3]$, using a single multiplication.

Note that in Example 4, although the online phase of the problem "evaluate $[\mathfrak{l}_1]^{12345}$ without square-and-multiply" was efficient, the pre-computation step generally runs in superpolynomial time, as it requires computing both discrete logarithms and lattice reduction.

1.3 Quaternion Algebras

Next, we extend our study of division algebras over $\mathbb{Q}$ to the non-commutative setting, by introducing the main protagonist of this thesis, namely quaternion algebras over $\mathbb{Q}$. We will see plenty of similarities with definite quaternion algebras, and imaginary quadratic number fields, as introduced in Section 1.2. The material in this section is largely based on the book by Voight [78].

Definition 1.3.1. A $\mathbb{Q}$-algebra B is a **quaternion algebra** (over $\mathbb{Q}$) if it admits a $\mathbb{Q}$-basis $1, \mathbf{i}, \mathbf{j}, \mathbf{k}$ such that

$$\mathbf{i}^2 = a, \qquad \mathbf{j}^2 = b, \qquad \mathbf{ij} = \mathbf{k} = -\mathbf{ji}$$

for some $a, b \in \mathbb{Q}^\times$.

We will sometimes use the notation $\left(\frac{a,b}{\mathbb{Q}}\right)$ for B to emphasise a, b as above. One simple example of a quaternion algebra is $\mathbf{M}_2(\mathbb{Q})$, the full 2-by-2 matrix ring over $\mathbb{Q}$. For instance, this can be checked by verifying that

$$\mathbf{i} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \qquad \mathbf{j} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

generates $\mathbf{M}_2(\mathbb{Q})$ as a $\mathbb{Q}$-algebra, and shows that $\mathbf{M}_2(\mathbb{Q}) \cong \left(\frac{1,1}{\mathbb{Q}}\right)$. However, note that $\mathbf{M}_2(\mathbb{Q})$ is in many ways very different from the quaternion algebras we will study.

We saw in Proposition 1.2.3 that the primes in $\mathbb{Z}$ which split in $\mathfrak{O}_K$ determined K up to isomorphism, as the product of the split primes was *almost* equal to $-d_K$. Defining the **places** of $\mathbb{Z}$ to be $\operatorname{Spec}\mathbb{Z} \cup \{\infty\}$, we can extend the notion of ramification to ∞ by defining K to be ramified at ∞ if K does not embed into $\mathbb{R}$, or equivalently, if

$$K \otimes \mathbb{R} = \mathbb{C}.$$

Then, d_K equals the product of the odd primes ramified in K, multiplied by 4 if K is ramified at 2, and multiplied by -1 if K is ramified at ∞.

We wish to extend this notion of ramification to B, which will turn out to be an extremely useful invariant, determining B up to isomorphism. It turns out that the correct generalisation is to study what happens when we look at the completion at places p, i.e.

$$B \otimes \mathbb{Q}_p$$

where $\mathbb{Q}_\infty = \mathbb{R}$. It turns out that there are essentially only two options for $B \otimes \mathbb{Q}_p$, either the full matrix ring $\mathbf{M}_2(\mathbb{Q}_p)$, or a division ring, and we say that B is ramified in the latter case, i.e. when $B \otimes \mathbb{Q}_p$ is a division ring.

Remark 3. *The reason that this generalises the concept of ramification can be understood as follows: We know that $B \otimes \mathbb{Q}_p$ has a unique maximal order $\mathcal{O}$, which equals the* ***valuation ring*** *of B, i.e. the elements of non-negative valuation. Further, $\mathcal{O}$ has a unique maximal two-sided ideal P, consisting of the elements of positive valuation. And* *further, one can show that $P^2 = p\mathcal{O}$ if and only if $B \otimes \mathbb{Q}_p$ is division. This indeed generalises ramification in a quadratic number field K; Writing $\mathfrak{O}_{K,p}$ for the valuation ring in $K \otimes \mathbb{Q}_p$ and $\mathfrak{P}$ for the unique maximal ideal in $\mathfrak{O}_{K,p}$, local field theory tells us that p is ramified in K if and only if $\mathfrak{P}^2 = p\mathfrak{O}_{K,p}$.*

Writing $\operatorname{Ram} B$ for the set of ramified places of B, we can now state the following theorem:

Theorem 1.3.2. *Let B and B' be two quaternion algebras over $\mathbb{Q}$. Then $B \cong B'$ if and only if $\operatorname{Ram} B = \operatorname{Ram} B'$.*

Proof. See [78, Proposition 14.3.1]. □

It is now clear that $B = \mathbf{M}_2(\mathbb{Q})$ is a very special example, namely the quaternion algebra ramified at no places. Further, one can show that any quaternion algebra is always ramified at a finite, even number of places. Thus, any squarefree number d determines a unique quaternion algebra B, by defining B to be the quaternion ramified at primes $p \mid d$, and ∞ if d has an odd number of prime factors. This invariant is called the **discriminant**, defined below.

Definition 1.3.3. Let B be a quaternion algebra over $\mathbb{Q}$. The **discriminant** $\operatorname{disc} B$ is defined as the product over all ramified primes of B.

We will be especially interested in the quaternion algebra ramified at p and ∞, for some prime p. This will be our main object of study, as this corresponds to the **endomorphism algebras** of supersingular elliptic curves (Theorem 2.2.5). We will denote this quaternion algebra by $B_{p,\infty}$.

Proposition 1.3.4. *Let $B = \left(\frac{a,b}{\mathbb{Q}}\right)$ be a quaternion algebra, and let p be an odd prime. Then, B is ramified at p and ∞ if $p \equiv 3 \pmod 4$ and*

$$a = -1 \quad \textit{and} \quad b = -p,$$

or if $p \equiv 1 \pmod 4$ and

$$a = -q \quad \textit{and} \quad b = -p \qquad \textit{where } q \equiv 3 \pmod 4 \textit{ is a prime satisfying } \left(\frac{q}{p}\right) = -1$$

Proof. Follows from the proof of [78, Proposition 14.2.7]. In particular, see [78, Example 14.2.13]. □

The above is not an if and only if, but it is very useful for constructing explicit representations of the quaternion algebra we are interested in.

1.3.1 Orders in Quaternion Algebras, and Their Ideals

Throughout this section, let B denote a quaternion algebra over $\mathbb{Q}$.

Recall that an **order** $\mathcal{O}$ is a subring of B that has rank 4 as a $\mathbb{Z}$-module. One construction of orders, which will come up time and time again throughout this thesis, is from the following definition.

Definition 1.3.5. Let I be a lattice in B. The **left order** of I (resp. the **right order** of I), denoted $\mathcal{O}_L(I)$ (resp. $\mathcal{O}_R(I)$), is defined as

$$\mathcal{O}_L(I) := \{\alpha \in B \mid \alpha I \subseteq I\}$$
$$(\text{resp. } \mathcal{O}_R(I) := \{\alpha \in B \mid I\alpha \subseteq I\}).$$

As suggested by the name, one can show that $\mathcal{O}_L(I)$ is in fact an order. Next, we define a special class of orders, which will be most important to us, as endomorphism rings of supersingular elliptic curves are examples of these, see Theorem 2.2.5.

Definition 1.3.6. Let $\mathcal{O} \subseteq B$ be an order. Further, $\mathcal{O}$ is called a **maximal** order if $\mathcal{O} \subseteq \mathcal{O}'$ implies $\mathcal{O} = \mathcal{O}'$ for any other order $\mathcal{O}' \subseteq B$.

In the commutative case, we saw that there exists a unique maximal order $\mathfrak{O}_K \subseteq K$ consisting of all the integral elements in K. The problem in the non-commutative case is that the set of all integral elements in B is no longer necessarily closed under multiplication, hence not an order! A consequence of this is that there can exist many maximal orders in B, instead of only one.

To determine whether an order is maximal, we introduce the notion of a discriminant. The general notion of discriminant exists for general lattices in simple algebras with an involution[4]. However, we will only use the special notion for orders in quaternion algebras.

Definition 1.3.7. Let A be a simple, finite-dimensional algebra over $\mathbb{Q}$ with an involution, and let $L \subseteq A$ be a lattice with basis $\alpha_1, \dots, \alpha_n$. Then the **discriminant** of L is defined as

$$\operatorname{disc} \mathcal{O} = \det(\operatorname{trd}(\alpha_i \overline{\alpha}_j)_{i,j \in \{1,\dots,n\}}),$$

and is independent of the choice of basis.

Further, if B is a quaternion algebra, and $\mathcal{O} \subset B$ is an order, then $\operatorname{disc} \mathcal{O}$ is always a square, and thus we can define the **reduced discriminant** of $\mathcal{O}$ to be

$$\operatorname{discrd} \mathcal{O} = \sqrt{\operatorname{disc} \mathcal{O}}.$$

The key properties of the reduced discriminant we need are the following.

Proposition 1.3.8. *Let B be a quaternion algebra, and let $\mathcal{O}' \subseteq \mathcal{O}$ be orders in B. Then*

[4] For instance, for an imaginary quadratic order $\mathfrak{O} = \mathbb{Z} + f\mathfrak{O}_K$, we have $\operatorname{disc} \mathfrak{O} = f^2 d_k$ as before.

(i) $\operatorname{discrd}\mathcal{O}' = [\mathcal{O}' : \mathcal{O}]\operatorname{discrd}\mathcal{O}$

(ii) $\operatorname{discrd}\mathcal{O} = \operatorname{disc} B$ *if and only if* $\mathcal{O}$ *is maximal.*

Proof. See [78, Lemma 15.2.15] for the first statement, and [78, Theorem 15.5.5] for the second statement. □

Paper 4 builds on a set of congruence conditions that can be deduced from (i) in Proposition 1.3.8.

Example 5: A maximal order. We work in the quaternion algebra

$$B_{p,\infty} = \left(\frac{-11, -109}{\mathbb{Q}}\right),$$

which can be confirmed to be ramified at $p = 109$ and ∞ by Proposition 1.3.4, thus we have $\operatorname{disc}(B) = 109$. Inside $B_{p,\infty}$, consider first the order

$$\mathcal{O}' = \alpha_1\mathbb{Z} + \alpha_2\mathbb{Z} + \alpha_3\mathbb{Z} + \alpha_4\mathbb{Z} = \mathbb{Z} + \frac{1+\mathbf{i}}{2}\mathbb{Z} + \mathbf{j}\mathbb{Z} + \frac{\mathbf{j}+\mathbf{k}}{2}\mathbb{Z} = \mathbb{Z}\langle\frac{1+\mathbf{i}}{2}, \mathbf{j}\rangle.$$

Using the basis above, we compute

$$\operatorname{trd}(\alpha_i\overline{\alpha}_j)_{i,j\in\{1,\dots,4\}} = \begin{pmatrix} 2 & 1 & 0 & 0 \\ 1 & 6 & 0 & 0 \\ 0 & 0 & 218 & 109 \\ 0 & 0 & 109 & 654 \end{pmatrix},$$

from which we compute the determinant

$$\operatorname{discrd}\mathcal{O}' = \sqrt{1437601} = 11 \cdot 109.$$

Thus, by Proposition 1.3.8, $\mathcal{O}'$ is not maximal. However, consider instead

$$\mathcal{O} = \beta_1\mathbb{Z} + \beta_2\mathbb{Z} + \beta_3\mathbb{Z} + \beta_4\mathbb{Z} = \mathbb{Z} + \frac{1+\mathbf{i}}{2}\mathbb{Z} + \frac{\mathbf{j}+\mathbf{k}}{2}\mathbb{Z} + \frac{\mathbf{i}-\mathbf{k}}{11}\mathbb{Z}.$$

It is straightforward to confirm that $\mathcal{O}' \subseteq \mathcal{O}$. Further, by again computing

$$\operatorname{discrd}\mathcal{O} = \sqrt{\operatorname{trd}(\beta_i\overline{\beta}_j)_{i,j\in\{1,\dots,4\}})} = \sqrt{\det\begin{pmatrix} 2 & 1 & 0 & 0 \\ 1 & 6 & 0 & 1 \\ 0 & 0 & 654 & 109 \\ 0 & 1 & 109 & 20 \end{pmatrix}} = 109,$$

we see that $\mathcal{O}$ is indeed maximal.

Next, we look at fractional ideals of orders in B. By definition, left fractional $\mathcal{O}$-ideals satisfy $\mathcal{O} \subseteq \mathcal{O}_L(I)$. This motivates the following definition.

Definition 1.3.9. Let I be a lattice in B, and $\mathcal{O} \subseteq B$ an order. Then, I is said to be a **proper** left fractional $\mathcal{O}$-ideal, resp. a **proper** right fractional $\mathcal{O}$-ideal, if

$$\mathcal{O} = \mathcal{O}_L(I),$$

resp.

$$\mathcal{O} = \mathcal{O}_R(I).$$

Indeed, as in the commutative case, we have defined fractional ideals, mainly to allow inverses of ideals. Still, we will prefer to work with ideals when we can, i.e. fractional ideals contained in their right or left order (sometimes referred to as **integral** ideals). One could wonder whether it is possible for an ideal to be left integral, but not right integral. Fortunately, the following proposition tells us that this never happens.

Proposition 1.3.10. *Let I be a lattice. Then the following are equivalent:*

(i) $I^2 \subseteq I$.

(ii) $I \subseteq \mathcal{O}_L(I)$, *i.e. I is a left $\mathcal{O}_L(I)$-ideal,*

(iii) $I \subseteq \mathcal{O}_R(I)$ *i.e. I is a right $\mathcal{O}_R(I)$-ideal.*

(iv) $I \subseteq \mathcal{O}_R(I) \cap \mathcal{O}_L(I)$.

Proof. See [78, Lemma 16.2.8]. □

Next, we start considering products of ideals. Due to the non-commutative nature of B, the product of ideals is usually not so well-behaved. The following definition gives the case in which it is.

Definition 1.3.11. Let $I, J \subseteq B$ be two lattices. We say that I is **compatible** with J if $\mathcal{O}_R(I) = \mathcal{O}_L(J)$.

Similar to Section 1.2, we extend the notion of reduced norm to ideals by defining

$$\mathrm{nrd}(I) = \gcd(\{\mathrm{nrd}(\alpha) \mid \alpha \in I\}),$$

where $\mathrm{nrd}(\alpha)$ is defined as in Section 1.1. The following proposition tells us that if we only consider the product of compatible ideals, then this product is well-behaved.

Proposition 1.3.12. *Let $I, J \subseteq B$ be two compatible ideals. Then the product IJ satisfies*

- $\mathcal{O}_L(I) \subseteq \mathcal{O}_L(IJ)$

- $\mathcal{O}_R(J) \subseteq \mathcal{O}_R(IJ)$
- $\operatorname{nrd}(IJ) = \operatorname{nrd}(I)\operatorname{nrd}(J)$.

Proof. For (i), just note that $\mathcal{O}_L(I)I \subseteq I$, and (ii) follows analogously on the right. Finally, (iii) is proven in [78, Lemma 16.3.7]. □

The main class of ideals we will be working with are the invertible ones.

Definition 1.3.13. Let I be a lattice in B. Then I is said to be **invertible** if there exists another lattice I^{-1} such that

$$II^{-1} = \mathcal{O}_L(I)$$

and

$$I^{-1}I = \mathcal{O}_R(I).$$

Further, if I is invertible, then

$$I^{-1} = \bar{I}\frac{1}{\operatorname{nrd}(I)}.$$

One could perhaps imagine ideals that were invertible on the left, but not on the right. However, the existence of a one-sided inverse turns out to be equivalent to the existence of a two-sided inverse.

When restricting to maximal orders $\mathcal{O}$, and their ideals (which will be our main focus point), the following proposition is analogous to the quadratic case (Proposition 1.2.4).

Proposition 1.3.14. *Let $\mathcal{O} \subseteq B$ be a maximal order, and let I be a left fractional $\mathcal{O}$-ideal. Then I is invertible, and $\mathcal{O}_R(I)$ is also maximal.*

Proof. See [78, Proposition 16.6.15]. □

1.3.2 The Class Set

Next, we want to consider some analogue of the material from Section 1.2.2. Unfortunately, we will not end up with a group structure on the set of equivalence classes of ideals, because elements here will typically not be compatible.

Throughout this section, we will fix B to be a definite quaternion algebra, and $\mathcal{O} \subseteq B$ to be an order. Again, many of the statements will remain true in more general settings, but this is the case of importance to us.

Definition 1.3.15. Let I, J be two left fractional $\mathcal{O}$-ideals. Then I, J are said to be **left equivalent** if $I = J\alpha$ for some $\alpha \in B^\times$.

Note that the definition above is analogous to ideals in the same equivalence class in the class group of an imaginary quadratic field. Hence we can make the following definition.

Definition 1.3.16. The **Class Set** of $\mathcal{O}$, denoted $\operatorname{Cls}\mathcal{O}$ is defined as

$$\operatorname{Cls}\mathcal{O} := \{I \subseteq B \mid I \text{ invertible left fractional } \mathcal{O}\text{-ideal}\}/\sim$$

where $I \sim J$ if and only if I and J are left equivalent.

The size of the class set is easy to compute given the factorisation of the discriminant[5], as it is a corollary of a deep theorem referred to as the Eichler mass formula.

Theorem 1.3.17 (Eichler's mass formula)**.** *Let B be a definite quaternion algebra with $\operatorname{disc} B = D$, and let $\mathcal{O} \subseteq B$ be a maximal order. Then*

$$\sum_{[J]\in\operatorname{Cls}\mathcal{O}} \frac{1}{\mathcal{O}_R(J)^\times} = \frac{\phi(D)}{24},$$

where ϕ denotes the Euler-phi function.

Proof. See [78, Theorem 25.3.15] for the case of maximal orders, or alternatively, the incredible generalisation that works for all orders [78, Main Theorem 25.3.19]. □

Unfortunately, there is no natural group structure on $\operatorname{Cls}\mathcal{O}$. For instance, given $[I], [J] \in \operatorname{Cls}\mathcal{O}$, their product IJ need not even be a left fractional $\mathcal{O}$-ideal.

Even if there is no group structure here, it will be worthwhile to study some aspects of this class set more closely. One point we will now elaborate on over the next few pages, is the following: In the case that we are most interested in, when $\mathcal{O}$ is a maximal order, we will see that $\operatorname{Cls}\mathcal{O}$ is *almost* in bijection with the isomorphism classes of maximal orders. The failure of this to be a bijection is something that will haunt us throughout this thesis.

Definition 1.3.18. Let $\mathcal{O}, \mathcal{O}' \subset B$ be two orders, such that there exists in ideal I satisfying $\mathcal{O}_L(I) = \mathcal{O}$, and $\mathcal{O}_R(I) = \mathcal{O}'$. Then I is said to be a **connecting** ideal.

In fact, when $\mathcal{O}, \mathcal{O}'$ are maximal, we can be even more specific.

Proposition 1.3.19. *Let $\mathcal{O}, \mathcal{O}' \subseteq B$ be maximal orders. Then*

$$I = \mathcal{O}\mathcal{O}'$$

is a connecting ideal between them.

In light of the above proposition, when given two maximal orders $\mathcal{O}, \mathcal{O}'$, we will sometimes refer to **the connecting ideal** between them as the ideal

$$I = N\mathcal{O}\mathcal{O}',$$

where $N \in \mathbb{Q}$ is such that I is a primitive, integral ideal.

[5] Which may be surprising, as computing the size of the class group of an imaginary quadratic order is hard!

Proposition 1.3.20. *Let $\mathcal{O}, \mathcal{O}' \subseteq B$ be two orders, with $\rho : \mathcal{O} \to \mathcal{O}'$, a ring-isomorphism. Then there exists some $\beta \in B^\times$ such that $\rho(\alpha) = \beta^{-1}\alpha\beta$ for all $\alpha \in \mathcal{O}$. In particular, $\mathcal{O}' = \beta^{-1}\mathcal{O}\beta$.*

Proof. This is a direct consequence of Skolem–Noether 1.1.4, by noting that ρ extends to a $\mathbb{Q}$-linear automorphism of B by tensoring with $\mathbb{Q}$. □

When two orders are isomorphic, they are said to be of the same type. By Proposition 1.3.20, two orders $\mathcal{O}, \mathcal{O}'$ of the same type satisfy $\mathcal{O}' = \beta^{-1}\mathcal{O}\beta$ for some $\beta \in B^\times$. One can then show that $I = \mathcal{O}\beta = \beta\mathcal{O}'$ is a connecting ideal, hence two orders of the same type are connected by a principal ideal.

However, there may also exist connecting ideals that are not principal. For the remainder of this subsection, we will limit ourselves to maximal orders $\mathcal{O}$, as otherwise the theory becomes rather complicated. For the general case, see Voight [78, Chapter 18].

Let $\mathcal{O}$ be a maximal order. We define $\mathrm{I}(\mathcal{O})$ to be the set of (necessarily invertible) two-sided fractional $\mathcal{O}$-ideals. As in the quadratic case, one can show that these ideals also have unique factorisation into prime ideals, and that they are all invertible, hence $\mathrm{I}(\mathcal{O})$ is again a free abelian group generated by the prime two-sided fractional $\mathcal{O}$-ideals.

Inside $\mathrm{I}(\mathcal{O})$, let $\mathrm{P}(\mathcal{O})$ denote the subgroup of principal two-sided fractional $\mathcal{O}$-ideals. In light of Definition 1.2.5, the quotient $\mathrm{I}(\mathcal{O})/\,\mathrm{P}(\mathcal{O})$ is practically begging to be studied. In the case of maximal orders, this group has a relatively simple, and predictable description (unlike in the quadratic case!).

Proposition 1.3.21. *Let $\mathcal{O} \subseteq B$ be a maximal order, and let $\operatorname{disc} B = D$. For every prime $p_i \mid D$, let P_i denote the unique two-sided ideal of norm p_i. Then the quotient group $\mathrm{I}(\mathcal{O})/\,\mathrm{P}(\mathcal{O})$ is generated by the P_i. Moreover,*

$$\mathrm{I}(\mathcal{O})/\,\mathrm{P}(\mathcal{O}) \cong (\mathbb{Z}/2\mathbb{Z})^r$$

where $0 \leq r \leq d$, where d is the number of prime factors of D.

Proof. By [78, Theorem 18.3.6] we have the exact sequence

$$0 \to \mathrm{I}(\mathbb{Z}) \to \mathrm{I}(\mathcal{O}) \to \prod_{p \mid D} \mathbb{Z}/2\mathbb{Z} \to 0,$$

obtained by sending $n\mathbb{Z} \in \mathrm{I}(\mathbb{Z})$ to $\mathcal{O}n\mathcal{O} \in \mathrm{I}(\mathcal{O})$. Further, from the proof of [78, Theorem 18.3.6], it is clear that the group $\mathrm{I}(\mathcal{O})/\,\mathrm{I}(\mathbb{Z})$ is generated by the ideals P_i. Thus, by noting that the image of $\mathrm{I}(\mathbb{Z})$ lies in $\mathrm{P}(\mathcal{O})$, the natural surjection from $\mathrm{I}(\mathcal{O})/\,\mathrm{I}(\mathbb{Z})$ to $\mathrm{I}(\mathcal{O})/\,\mathrm{P}(\mathcal{O})$ gives the desired result. □

Warning 4. *Let O be an order. When O is an imaginary quadratic order, $\mathrm{I}(O)/\,\mathrm{P}(O)$ is naturally isomorphic to the Picard group $\operatorname{Pic} O$, which is defined differently [78, Definition 18.4.1]. However, in general, this is* not *true, and for quaternion orders, these notions may or may not be the same. In fact, when O is not maximal, $\mathrm{P}(O)$ need not even be a normal subgroup of $\mathrm{I}(O)$, hence the "group" $\mathrm{I}(O)/\,\mathrm{P}(O)$ may be nonsensical.*

In our case of interest, we have $\operatorname{disc} B_{p,\infty} = p$, and $\mathrm{I}(\mathcal{O})/\mathrm{P}(\mathcal{O})$ is either trivial or isomorphic to $\mathbb{Z}/2\mathbb{Z}$, with the former case happening if and only if the unique two-sided ideal of norm p is principal. The natural question of when this happens will be given a particularly interesting answer in Chapter 3; It turns out to be exactly when $\mathcal{O}$ is the endomorphism ring of an elliptic curve defined over $\mathbb{F}_p$.

For now, we will be satisfied by describing the difference between the set $\operatorname{Cls}\mathcal{O}$ and the set of isomorphism classes of maximal orders in B, denoted $\operatorname{Type}\mathcal{O}$.[6]

Proposition 1.3.22. *Let $\mathcal{O}$ be a maximal order. The map*

$$\begin{aligned} f : \operatorname{Cls}\mathcal{O} &\to \operatorname{Type}\mathcal{O}, \\ f([I]) &= [\mathcal{O}_R(I)] \end{aligned}$$

is a surjective map of sets. Further, we have that

$$\# f^{-1}(\mathcal{O}') = \# \mathrm{I}(\mathcal{O}')/\mathrm{P}(\mathcal{O}').$$

Proof. First, the map f is well defined: If $[I] = [J]$, then $I = J\alpha$ for some $\alpha \in B^\times$, hence $\mathcal{O}_R(J) = \alpha^{-1}\mathcal{O}_R(I)\alpha$. Further, by Proposition 1.3.19, the map is surjective, as given a maximal order $\mathcal{O}'$, we have that $\mathcal{O}_R(\mathcal{O}\mathcal{O}') = \mathcal{O}'$.

Second, given a maximal order $\mathcal{O}'$, the map

$$\begin{aligned} \mathrm{I}(\mathcal{O}')/\mathrm{P}(\mathcal{O}') &\to f^{-1}(\mathcal{O}') \subseteq \operatorname{Cls}\mathcal{O}, \\ [J] &\to [\mathcal{O}\mathcal{O}'J] \end{aligned}$$

is a bijection [78, Proposition 18.5.10] of finite sets, finishing the proof. □

Throughout this thesis, we will typically work with $\operatorname{Cls}\mathcal{O}$, and not $\operatorname{Type}\mathcal{O}$, see for instance Definition 1.4.4.

1.3.3 Ternary and Quartic Quadratic Forms.

Analogously to Section 1.2.3, we can associate ideals in quaternion algebras to quartic quadratic forms. It should come as no surprise that we do not have anything analogous to the composition laws, however, we will still see that this association is useful in the quaternion context too.

Given an integral ideal I with a fixed $\mathbb{Z}$-basis $\alpha_1, \alpha_2, \alpha_3, \alpha_4$, we define the "norm form" of I to be the quartic quadratic form

$$f(x, y, z, w) = \operatorname{nrd}(x\alpha_1 + y\alpha_2 + z\alpha_3 + w\alpha_4),$$

[6] For general orders $\mathcal{O} \subset B$, $\operatorname{Type}\mathcal{O}$ refers to the isomorphism classes of orders connected to $\mathcal{O}$, but by Proposition 1.3.19 all maximal orders are connected.

which is again well defined up to a change of basis of I. In many cases, it can also be more useful to work with the normalized map[7]

$$\begin{aligned} q_I &: I \to \mathbb{Z}, \\ q_I(\alpha) &= \mathrm{nrd}(\alpha)/\mathrm{nrd}(I). \end{aligned}$$

One essential use case of this is the following Lemma, which shows that finding equivalent ideals of a given norm corresponds to finding integral representations of the norm form.

Lemma 1.3.23. *Let $I \subset B$ be an integral ideal. The map*

$$\begin{aligned} \chi_I &: I \to \{J \subset B \mid J \textit{ integral, and (left) equivalent to } I\}, \\ \chi_I(\alpha) &= I\frac{\bar{\alpha}}{\mathrm{nrd}(I)} \end{aligned}$$

is well defined and surjective. Further, the ideal $\chi_I(\alpha)$ has reduced norm $q_I(\alpha)$.

Proof. See [48, Lemma 5] for well definiteness and [30, Lemma 1] for surjectivity. The statement on the norm follows from the multiplicativity of the norm. □

The KLPT algorithm [48], and its related algorithms, which are of huge importance to this thesis and isogeny-based cryptography in general (see Section 3.3.1), are algorithms for solving exactly such norm forms.

In the special case of being given an order $\mathcal{O} \subseteq B$, one can also instead associate a ternary quadratic form. To do this, consider the $\mathbb{Z}$-submodule $\mathcal{O}^{(0)} \subseteq \mathcal{O}$ defined as

$$\mathcal{O}^{(0)} = \{\alpha \in \mathcal{O} \mid \mathrm{trd}(\alpha) = 0\}.$$

It can be shown (e.g. explicitly by using a basis of $\mathcal{O}$ in HNF form) that this is a $\mathbb{Z}$-module of rank 3 contained in B. Hence, the associated (ternary) quadratic form of $\mathcal{O}$ can again be defined as

$$g(x, y, z) = \mathrm{nrd}(x\beta_1 + y\beta_2 + z\beta_3),$$

where $\beta_1, \beta_2, \beta_3$ is a basis of $\mathcal{O}^{(0)}$. While such ternary quadratic forms seem to be a special case, it turns out that, perhaps surprisingly, there is an exact[8] correspondence between integral ternary quadratic forms and quaternion orders over $\mathbb{Z}$ [78, Main Theorem 22.1.1.].

In Paper 4, this association will be important to us in relation to **optimal embeddings** (see Definition 1.4.1), as computing optimal embeddings into a given quaternion

[7] Notice the similarity with the association in Theorem 1.2.9, though we seem to have lost a sign flip on the way. One explanation for this is that we were previously considering things up to *proper* equivalence, and the sign flip corresponds to an *improper* equivalence, i.e. by a change of basis given by $M = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \in \mathbf{GL}_2(\mathbb{Z}) \setminus \mathbf{SL}_2(\mathbb{Z})$.

[8] Up to appropriate notions of "isomorphism".

order (referred to as the **order embedding problem**) correspond to finding representations of its associated ternary quadratic form. Note also that it is clear from the association above, that solving the order embedding problem in full generality is equivalent to finding integral representations of ternary quadratic forms.

1.4 Optimal Embeddings

We have seen that there are several analogues between the (integral) theory of definite quaternion algebras and imaginary quadratic number fields. In this section, we briefly study a very explicit connection between these two, namely the theory of optimal embeddings.

In this section, let K be an imaginary number field and let B be a definite quaternion algebra, such that there exists an embedding $\iota : K \hookrightarrow B$.

We are mainly interested in the integral theory, coming from embeddings

$$\epsilon : \mathfrak{O} \to \mathcal{O},$$

where $\mathfrak{O}, \mathcal{O}$ are orders in K and B respectively. Note that ϵ can be extended (uniquely) to an embedding $\iota : K \hookrightarrow B$ such that $\iota \mid_{\mathfrak{O}} = \epsilon$, hence we consider all such ϵ to be the restriction of an embedding $\iota : K \hookrightarrow B$.

Definition 1.4.1. Let $\mathfrak{O} \subseteq K$ be an order in K, and let $\mathcal{O} \subseteq B$ be an order in B. An embedding

$$\iota \mid_{\mathfrak{O}} : \mathfrak{O} \hookrightarrow \mathcal{O}$$

is an **optimal embedding** if

$$\iota(K) \cap \mathcal{O} = \iota(\mathfrak{O}).$$

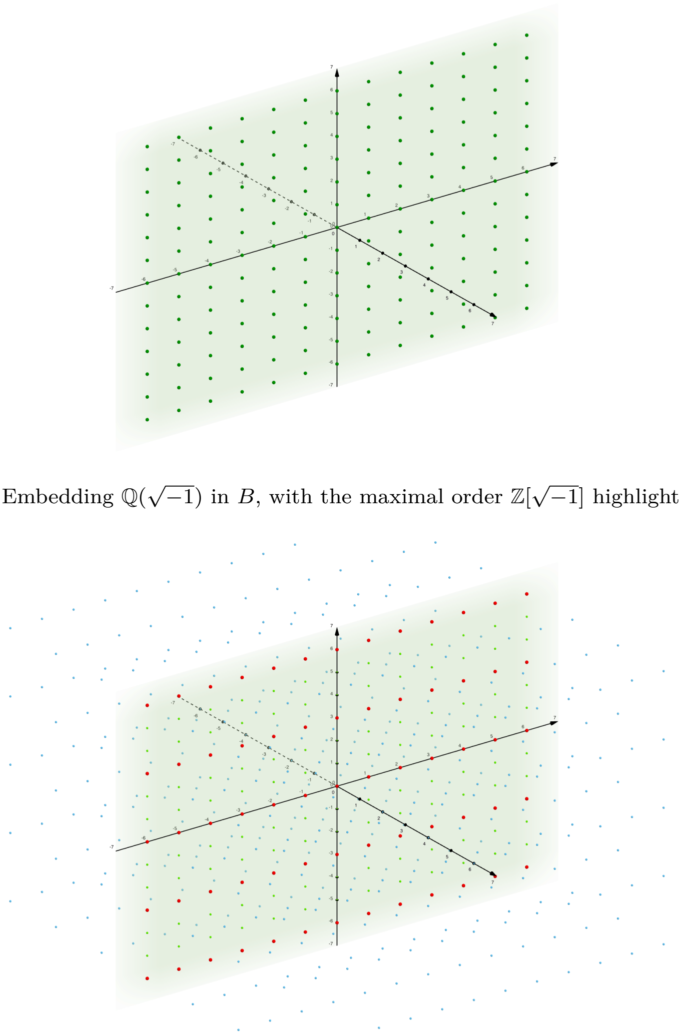
Writing $\mathfrak{O} = \mathbb{Z}[\omega]$ for some generator $\omega \in \mathfrak{O}$, we see that finding an optimal embedding $\iota : \mathfrak{O} \hookrightarrow \mathcal{O}$ amounts to finding an element $\alpha \in \mathcal{O}$ such that $\mathrm{n}(\omega) = \mathrm{nrd}(\alpha), \mathrm{t}(\omega) = \mathrm{trd}(\alpha)$. Assume for simplicity that ω has $\mathrm{t}(\omega) = 0$. Then it is easy to see that finding the embedding is the same as solving the ternary quadratic form related to $\mathcal{O}$, defined in Section 1.3.3.

We also introduce the following notation.

Definition 1.4.2. Let $\iota \mid_{\mathfrak{O}} \mathfrak{O} \to \mathcal{O}$ be an embedding. We define the pair $(\mathcal{O}, \iota)$ to be an **$\mathfrak{O}$-oriented order**. Further, if ι is optimal, $(\mathcal{O}, \iota)$ is said to be a **primitively $\mathfrak{O}$-oriented order**.

We have looked at isomorphisms between orders in B, however, when looking at oriented orders, we want to restrict to those isomorphisms that preserve the orientation.

Definition 1.4.3. Let $(\mathcal{O}, \iota), (\mathcal{O}', \iota')$ be two primitively oriented orders. An **oriented isomorphism** is an isomorphism $\rho : \mathcal{O} \to \mathcal{O}'$ such that $\rho \circ \iota = \iota'$.



(a) Embedding $\mathbb{Q}(\sqrt{-1})$ in B, with the maximal order $\mathbb{Z}[\sqrt{-1}]$ highlighted.

(b) The induced embedding is optimal for $\mathbb{Z}[3\sqrt{-1}]$ in a quaternion order.

Figure 1.2: Figure showing an embedding of $\iota : \mathbb{Q}(\sqrt{-1}) \hookrightarrow B$. The green dots show $\iota(\mathbb{Z}[\sqrt{-1}])$, the blue dots show a quaternion order $\mathcal{O}$, and the red dots show the intersection $\iota(K) \cap \mathcal{O} = \iota(\mathbb{Z}[3\sqrt{-1}])$. Note that the picture is for illustrative purposes, and clearly inaccurate as the "quaternion algebra" is only 3-dimensional. This is because drawing the projection of 4-dimensional space on a 2-dimensional medium (this sheet of paper) generally leads to everyone having a bad time.

We want to count the number of embeddings into a given order, up to oriented isomorphism. By the Skolem–Noether theorem (Theorem 1.1.4), any isomorphism ρ of orders is given by conjugation by some element α, hence one can show that it is enough to consider the set

$$\operatorname{Emb}(\mathfrak{O}, \mathcal{O}, \mathcal{O}^\times) := \{(\mathcal{O}, \iota) \text{ primitively } \mathfrak{O}\text{-oriented order}\}/\sim,$$

where $\sim$ is the equivalence relation defined by conjugation by $\alpha \in \mathcal{O}^\times$. See [78, 30.3] for details.

Next, we will define a set which we will study in detail later. For this, we introduce the following notation: Consider a quadratic order $\mathfrak{O} \subset K$, and a fixed generator $\delta \in K$. For a given $\omega \in B$ satisfying $\operatorname{nrd}(\omega) = \operatorname{n}(\delta)$ and $\operatorname{trd}(\omega) = \operatorname{t}(\delta)$, we define the embedding

$$\iota_\omega : K \hookrightarrow B$$

to be the embedding defined by $\iota(\delta) = \omega$.

Definition 1.4.4. We define the **set of primitively $\mathfrak{O}$-oriented ideal classes** to be

$$\operatorname{Cls}_{\mathfrak{O}}(\mathcal{O}) := \{(I, \omega) \mid [I] \in \operatorname{Cls} \mathcal{O}, (\mathcal{O}_R(I), \iota_\omega) \text{ is a primitively oriented order}\}/\sim$$

where $(I, \omega) \sim (J, \omega')$ whenever there exists $\alpha \in B$ such that $I\alpha = J$ and $\alpha^{-1}\omega\alpha = \omega'$.

The definition above may seem unintuitive at first, but it is is a consequence of Proposition 1.3.22. In fact, we can naturally extend Proposition 1.3.22 to the oriented case.

Corollary 1.4.5. *Let* $\operatorname{Type}_{\mathfrak{O}}(\mathcal{O})$ *be the set*

$$\operatorname{Type}_{\mathfrak{O}}(\mathcal{O}) : \{(\mathcal{O}, \iota) \mid (\mathcal{O}', \iota) \textit{ is a primitively oriented order}\}/\sim$$

where $\sim$ is the equivalence relation defined by being isomorphic as primitively oriented orders. Then there is a natural surjection

$$\begin{aligned} g : \operatorname{Cls}_{\mathfrak{O}}(\mathcal{O}) &\to \operatorname{Type}_{\mathfrak{O}}(\mathcal{O}) \\ g(I, \omega) &= (\mathcal{O}_R(I), \iota_\omega). \end{aligned}$$

Further, we have that

$$\# g^{-1}(\mathcal{O}', \iota) = \# \operatorname{I}(\mathcal{O}')/\operatorname{P}(\mathcal{O}')$$

Proof. The map is well-defined. Let $(I, \omega), (J, \omega')$ be elements of the same class in $\operatorname{Cls}_{\mathfrak{O}}(\mathcal{O})$. Then, by definition, there exists an $\alpha \in B$ defining an oriented isomorphism between $(\mathcal{O}_R(I), \iota_\omega)$ and $(\mathcal{O}_R(J), \iota_{\omega'})$. For the second part, let $(I, \omega) \in \operatorname{Cls}_{\mathfrak{O}}(\mathcal{O})$. By Proposition 1.3.22, there are $\# \operatorname{I}(\mathcal{O}')/\operatorname{P}(\mathcal{O}')$ choices of ideal classes $[J]$ such that $\mathcal{O}_R(I) \cong \mathcal{O}_R(J)$, and the isomorphism induces bijections between the orientations. □

Computing the size of $\operatorname{Cls}_{\mathfrak{O}}(\mathcal{O})$ is a classical problem, and the theory of optimal embeddings has interesting results and applications. We restate one special case of a more general theorem [78, Theorem 30.7.3], which is our main interest.

Proposition 1.4.6. *Let $\mathfrak{O} \subseteq K$ be an order. Then*

$$\#\operatorname{Cls}_{\mathfrak{O}}(\mathcal{O}) = \#\operatorname{Cl}(\mathfrak{O})\left(1 - \left(\frac{d_K}{p}\right)\right).$$

Proof. Since equivalent ideals have isomorphic right orders, we can choose a representative $[I]$ for each ideal class, and compute all the orientations of that order, up to oriented isomorphism, i.e. up to conjugation by $\mathcal{O}_R(I)^\times$. Thus, we see that

$$\#\operatorname{Cls}_{\mathfrak{O}}(\mathcal{O}) = \sum_{[I]\in\operatorname{Cls}\mathcal{O}} \#\mathrm{Emb}(\mathfrak{O}, \mathcal{O}_R(I), \mathcal{O}_R(I)^\times),$$

which is a special case of [78, Theorem 30.7.3]. □

Proposition 1.4.6 says that the size of $\operatorname{Cls}_{\mathfrak{O}}(\mathcal{O})$ is either $0, h(\mathfrak{O})$ or $2h(\mathfrak{O})$, depending only on the Legendre symbol $\left(\frac{d_K}{p}\right)$. In particular, it does not depend on $\mathcal{O}$. So even if finding the individual $\#\mathrm{Emb}(\mathfrak{O}, \mathcal{O}_R(I), \mathcal{O}_R(I)^\times)$ seems to be a hard problem in general, we know the total number of solutions when summing over the right orders of (a representative of) each ideal class of any maximal order in $B_{p,\infty}$.

Example 6: Oriented ideal classes. Let $p = 109$, and let $B_{p,\infty} = (-11, -109 \mid \mathbb{Q})$. Since $p \equiv 1 \pmod{12}$, all the maximal orders have $\mathcal{O}^\times = \{\pm 1\}$, hence no inner automorphisms, making things a little easier (we can ignore conjugation). We fix the maximal order

$$\mathcal{O}_0 = \mathbb{Z} + \frac{1+\mathbf{i}}{2}\mathbb{Z} + \frac{\mathbf{j}+\mathbf{k}}{2}\mathbb{Z} + \frac{\mathbf{i}-\mathbf{k}}{11}\mathbb{Z}$$

from Example 5, and we compute a representative for each class $[I_i] \in \operatorname{Cls}\mathcal{O}_0$:

$$\begin{aligned}
I_0 &= \mathbb{Z}\mathbb{Z} + \frac{1+\mathbf{i}}{2}\mathbb{Z} + \mathbf{j}\mathbb{Z} + \frac{11+\mathbf{i}+11\mathbf{j}+\mathbf{k}}{22}\mathbb{Z}\\
I_1 &= 2\mathbb{Z} + (1+\mathbf{i})\mathbb{Z} + \frac{1+\mathbf{i}+2\mathbf{j}}{2}\mathbb{Z} + \frac{22+12\mathbf{i}+11\mathbf{j}+\mathbf{k}}{22}\mathbb{Z}\\
I_2 &= 2\mathbb{Z} + (1+\mathbf{i})\mathbb{Z} + \frac{3+\mathbf{i}+2\mathbf{j}}{2}\mathbb{Z} + \frac{33+\mathbf{i}+11\mathbf{j}+\mathbf{k}}{22}\mathbb{Z}\\
I_3 &= 2\mathbb{Z} + (1+\mathbf{i})\mathbb{Z} + (1+\mathbf{j})\mathbb{Z} + \frac{11+\mathbf{i}+11\mathbf{j}+\mathbf{k}}{22}\mathbb{Z}\\
I_4 &= 4\mathbb{Z} + (2+2\mathbf{i})\mathbb{Z} + \frac{1+\mathbf{i}+2\mathbf{j}}{2}\mathbb{Z} + \frac{44+34\mathbf{i}+11\mathbf{j}+\mathbf{k}}{22}\mathbb{Z}\\
I_5 &= 4\mathbb{Z} + (2+2\mathbf{i})\mathbb{Z} + \frac{3+3\mathbf{i}+2\mathbf{j}}{2}\mathbb{Z} + \frac{34\mathbf{i}+11\mathbf{j}+\mathbf{k}}{22}\mathbb{Z}\\
I_6 &= 8\mathbb{Z} + (4+4\mathbf{i})\mathbb{Z} + (7+\mathbf{i}+2\mathbf{j})\mathbb{Z} + \frac{33+\mathbf{i}+11\mathbf{j}+\mathbf{k}}{11}\mathbb{Z}
\end{aligned}$$

$$I_7 = 4\mathbb{Z} + (2+2\mathbf{i})\mathbb{Z} + (2+\mathbf{i}+\mathbf{j})\mathbb{Z} + \frac{55+\mathbf{i}+11\mathbf{j}+\mathbf{k}}{22}\mathbb{Z}$$
$$I_8 = 4\mathbb{Z} + (2+2\mathbf{i})\mathbb{Z} + (\mathbf{i}+\mathbf{j})\mathbb{Z} + \frac{33+23\mathbf{i}+11\mathbf{j}+\mathbf{k}}{22}\mathbb{Z}$$

We now first consider primitive orientations $\mathfrak{O} = \mathbb{Z}[\frac{1+\sqrt{-23}}{2}]$. We have that $h(\mathfrak{O}) = 3$, and $\left(\frac{11}{109}\right) = -1$, hence we expect $\mathrm{Cls}_{\mathfrak{O}}(\mathcal{O}_0)$ to contain 6 elements. Sure enough, we find representatives for each class in $\mathrm{Cls}_{\mathfrak{O}}(\mathcal{O})$ as follows:

$$\begin{aligned}\mathrm{Cls}_{\mathbb{Z}[\frac{1+\sqrt{-23}}{2}]}(\mathcal{O}_0) = \Big\{ &\left(I_3, \frac{11+12\mathbf{i}+\mathbf{k}}{22}\right), \left(I_3, \frac{11-12\mathbf{i}-\mathbf{k}}{22}\right), \\ &\left(I_7, \frac{11+12\mathbf{i}+\mathbf{k}}{22}\right), \left(I_7, \frac{11-12\mathbf{i}-\mathbf{k}}{22}\right), \\ &\left(I_8, \frac{11+12\mathbf{i}+\mathbf{k}}{22}\right), \left(I_8, \frac{11-12\mathbf{i}-\mathbf{k}}{22}\right)\Big\}\end{aligned}$$

We can also demonstrate the surjection onto $\mathrm{Type}_{\mathfrak{O}}(\mathcal{O}_0)$ given in Proposition 1.4.5. By conjugating by $\mathbf{j}$, we see that

$$\begin{aligned}\left(\mathcal{O}_R(I_7), \frac{11+12\mathbf{i}+\mathbf{k}}{22}\right) &\cong \left(\mathcal{O}_R(I_8), \frac{11-12\mathbf{i}-\mathbf{k}}{22}\right) \\ \left(\mathcal{O}_R(I_7), \frac{11-12\mathbf{i}-\mathbf{k}}{22}\right) &\cong \left(\mathcal{O}_R(I_8), \frac{11+12\mathbf{i}+\mathbf{k}}{22}\right).\end{aligned}$$

Next, we consider $\mathbb{Z}[\sqrt{-109}]$. In this case, we have that $h(\mathfrak{O}) = 6$, and $\left(\frac{109}{109}\right) = 0$, and thus we again expect $\mathrm{Cls}_{\mathbb{Z}[\sqrt{-109}]}(\mathcal{O}_0)$ to contain 6 elements. We find those 6 elements to be:

$$\begin{aligned}\mathrm{Cls}_{\mathbb{Z}[\sqrt{-109}}(\mathcal{O}_0) = \{ &(I_0, \mathbf{j}), (I_0, -\mathbf{j}), \\ &(I_3, \mathbf{j}), (I_3, -\mathbf{j}), \\ &\left(I_4, \frac{218+33\mathbf{j}-13\mathbf{k}}{88}\right), \left(I_4, \frac{-218-33\mathbf{j}+13\mathbf{k}}{88}\right)\Big\}\end{aligned}$$

This time, there turns out to be a bijection between $\mathrm{Cls}_{\mathbb{Z}[\sqrt{-109}]}(\mathcal{O}_0)$ and $\mathrm{Type}_{\mathfrak{O}}(\mathcal{O}_0)$. This has a natural explanation: The element giving the embedding of $\mathbb{Z}[\sqrt{-109}]$ generates a two-sided ideal of norm p, hence by the proof of Proposition 1.3.21, it is clear that these orders are precicely the orders that satisfy $\mathrm{I}(\mathcal{O})/\,\mathrm{P}(\mathcal{O}) \simeq \{1\}$.

Chapter 2

Elliptic Curves

We now shift our focus from the arithmetic world of central simple $\mathbb{Q}$-algebras, to the geometric world of elliptic curves. We will soon focus on **supersingular** elliptic curves, and their endomorphism rings before we in the next chapter connect these curves with the material in Chapter 1. For a more thorough introduction to this rich subject, see Silverman [71].

2.1 Elliptic Curves and Isogenies

In this thesis, our main focus will be on elliptic curves defined over large characteristic. So, from this point forward let K denote a field with $\mathrm{char}(K) \neq 2, 3$.

The general definition of an elliptic curve is a smooth projective curve of genus 1, with a specified basepoint $\infty \in E$. However, it is a standard fact that every elliptic curve (E, ∞) embeds into $\mathbb{P}^2$, through a map sending ∞ to $(0 : 1 : 0)$, and E to the curve defined by a **short Weierstrass equation**

$$Y^2Z = X^3 + AXZ^2 + BZ^3, \qquad A, B \in K,$$

hence, without loss of generality, it suffices to consider elliptic curves given by Weierstrass equations.[1] In this case, we also simply refer to elliptic curves as E, with the base point being understood to be $\infty = (0 : 1 : 0)$. As is common practice, when working with Weierstrass curves, we often ease notation by working with the affine coordinates $x = X/Z$ and $y = Y/Z$ at $Z \neq 0$, in which case there is a unique point at infinity, namely the specified basepoint.

However, note that we do not exclusively work with short Weierstrass equations. For instance, in some applications, where computational efficiency is the crux, one can

[1] To also cover $\mathrm{char}(K) = 2$ or 3, we need the longer equation $Y^2Z + a_1XYZ + a_3YZ^3 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3$, called a **long Weierstrass equation**.

often use different curve models. As an example of this, we work with the so-called **Montgomery model**

$$Y^2Z = X^3 + AX^2Z + XZ^2$$

in Chapter 3. For elliptic curves given by a Montgomery model, the basepoint is again understood to be $\infty = (0 : 1 : 0)$. However, note that not every curve has a Montgomery model over the same field.

As in the case of binary quadratic forms (Section 1.2.3), we can define a very explicit group law on the set of points of an elliptic curve.

Theorem 2.1.1. *Let E/K be an elliptic curve given by the Weierstrass equation*

$$E : y^2 = x^3 + Ax + B,$$

and let $P = (x_0, y_0), Q = (x_1, y_1)$ be two points on E, not equal to ∞.

- *If $x_0 = x_1$ and $y_0 = -y_1$, set $P + Q = \infty$.*
- *Otherwise, define*

$$\lambda = \begin{cases} \frac{y_0 - y_1}{x_0 - x_1} & \text{if } x_0 \neq x_1, \\ \frac{3x_0^2 + A}{2y_0} & \text{otherwise.} \end{cases}$$

  and set $x_2 := \lambda^2 - x_0 - x_1$, and define

$$P + Q := (x_2, \lambda(x_0 - x_2) - y_0).$$

Then, these addition laws turn E into an abelian group, with identity element ∞. Further, for any field $L \supseteq K$, the ***set of L-rational points****, denoted $E(L)$, is a subgroup.*

Proof. See [71, Proposition 2.2]. □

Using the notion of K-rational points, we can think of the group of points on E as a shorthand notation for $E(\bar{K})$.

Example 7: Rational points. Let $p = 31$, and consider the elliptic curve

$$E/\mathbb{F}_p : y^2 = x^3 + 29x + 5.$$

Simply by checking all possible values of x in $\mathbb{F}_p$, we see that

$$\begin{aligned} E(\mathbb{F}_p) = \{&\infty, (0,6), (0,25), (1,2), (1,29), (2,3), (2,28), (8,6), (8,25), (12,2), (12,29), \\ &(15,8), (15,23), (16,15), (16,16), (18,2), (18,29), (22,10), (22,21), (23,6), \\ &(23,25), (25,7), (25,24), (26,13), (26,18), (29,1), (29,30)\} \end{aligned}$$

Thus, we know that $E(\mathbb{F}_p)$ is an abelian group of order 27. Further, by using the addition laws in Theorem 2.1.1, we can verify that e.g. $P = (18, 29)$ has order 9, and that there are at least four points of order 3, thus the only possibility for $E(\mathbb{F}_p)$ is

$$E(\mathbb{F}_p) \cong \mathbb{Z}/9\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}.$$

Once we now know that any elliptic curve comes equipped with an abelian group structure, we can study two different types of points: torsion points, and points of infinite order. In this thesis, our focus is on elliptic curves over finite fields, hence we only have the former.

Given any abelian group G, and integer m, one defines the **m-torsion subgroup** as

$$G[m] = \{g \in G \mid mg = 0_G\}.$$

For elliptic curves, the structure of the m-torsion subgroup is captured in the following proposition:

Proposition 2.1.2. *Let E/K be an elliptic curve, and let $0 \neq m \in \mathbb{Z}$. Then, assuming* $\mathrm{char}(K) \nmid m$*, we have that*

$$E[m] = \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}.$$

Otherwise, if $m = p^e$ for $p =$ $\mathrm{char}(K)$*, we have that either*

$$E[p^e] = \mathbb{Z}/p^e\mathbb{Z}, \quad \textit{or} \quad E[p^e] = \{\infty\}.$$

Proof. See [71, Corollary 6.4]. □

Example 8: Full N-torsion subgroups. We continue from Example 7, letting E and p be as before. From the structure of the group $E(\mathbb{F}_p)$ we know that

$$\begin{aligned} E(\mathbb{F}_p)[3] = \{&\infty, (15,8), (15,23), (23,6), (23,25), \\ &(26,13), (26,18), (29,1), (29,30)\} \cong \mathbb{Z}/3\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}. \end{aligned}$$

Thus, by Proposition 2.1.2, we know that $E[3] = E(\mathbb{F}_p)[3]$, i.e. every 3-torsion point on E is $\mathbb{F}_p$-rational. However, we also know that

$$E(\mathbb{F}_p)[9] = \langle (18,29) \rangle \cong \mathbb{Z}/9\mathbb{Z}.$$

Thus, we know that we need to consider E over a larger extension to see all of $E[9]$. And sure enough, let $\mathbb{F}_{p^3} = \mathbb{F}_p(\omega)$ where $\omega^3 + \omega + 28 = 0$. We find that

$$E[9] = \langle (18,29), (22\omega^2 + \omega + 26, 17\omega^2 + 25\omega + 16) \rangle \subset E(\mathbb{F}_{p^3}).$$

Isogenies. Next, we wish to consider maps between elliptic curves. To do this, we start with the following definition

Definition 2.1.3. Let E/K be an elliptic curve given by the Weierstrass equation

$$E/K : y^2 = x^3 + Ax + B.$$

The **function field** of E, denoted $K(E)$ is the ring of rational functions on E. Explicitly, we have

$$K(E) \cong K(x, y)/\langle y^2 - x^3 - Ax - B\rangle.$$

While the isomorphism above may seem counter-intuitive, since E is really a projective curve, and $K(E)$ seems to depend on the choice of affine chart of E, one can show that this is independent of this choice.

Of course, as projective varieties, we can have morphisms between elliptic curves, but as we have seen, elliptic curves are also abelian groups, hence we want to restrict to morphisms preserving the group structure. It turns out that the correct definition is captured by the notion of an **isogeny**.

Definition 2.1.4. Let E, E' be two elliptic curves. A **morphism** $\varphi : E \to E'$ is a rational map

$$\begin{gathered}\varphi : E \to E',\\ \varphi((X : Y : Z)) = (f_0(X, Y, Z) : f_1(X, Y, Z) : f_2(X, Y, Z)), \quad f_0, f_1, f_2 \in K(E),\end{gathered}$$

that is everywhere regular, i.e. for every point $P \in E$, there exists a function $g \in \bar{K}(E)$ such that $gf_i(P)$ is defined and non-zero for at least one i.

An **isogeny** is a non-constant morphism φ that preserves the basepoint, i.e. satisfies $\varphi(\infty_E) = \infty_{E'}$.

Note that the definition above considers elements $f_0, f_1, f_2 \in K(E)$ as rational functions on E, not using the association $K(E) \cong K(x, y)/\langle y^2 - x^3 - Ax - B\rangle$. Still, in practice, we write the isogenies in affine coordinates (using the "usual" affine chart for curves in Weierstrass form), understanding that points for which the evaluation is undefined are points that map to $\infty_{E'}$.

It turns out that the relatively mild assumption on an isogeny is enough to induce group homomorphisms.

Theorem 2.1.5. *Let $\varphi : E \to E'$ be an isogeny defined over K. Then*

$$\varphi : E(L) \to E'(L)$$

is a group homomorphism for all $L \supseteq K$.

Proof. See [71, Theorem 4.8]. □

As with all morphisms (between projective varieties), an isogeny $\varphi : E \to E'$ induces an injection of function fields via the pullback

$$\begin{aligned} \varphi^* : \bar{K}(E') &\hookrightarrow \bar{K}(E), \\ \varphi^*(f) &= f \circ \varphi. \end{aligned}$$

This motivates the following definitions.

Definition 2.1.6. Let $\varphi : E \to E'$ be an isogeny. The **degree** of φ, denoted by $\deg \varphi$, is defined to be the degree of the field extension $\bar{K}(E)/\varphi^*\bar{K}(E')$. Further, φ is said to be **separable/inseparable/purely inseparable** if the field extension $\bar{K}(E)/\varphi^*\bar{K}(E')$ is separable/inseparable/purely inseparable respectively.

An **isomorphism** between elliptic curves is an isogeny of degree 1. Isomorphic elliptic curves therefore have isomorphic function fields, and typically, we will only work with elliptic curves up to isomorphism. Therefore, it is useful to have an invariant of an elliptic curve that is invariant in its isomorphism class. Given a curve in Weierstrass from

$$E : y^2 = x^3 + Ax + B,$$

define the ***j*-invariant** to be

$$j(E) = 1728 \frac{4A^3}{4A^3 + 27B^2}.$$

Then we have the following proposition.

Proposition 2.1.7. *Given two elliptic curves E, E' defined over K, we have that*

$$E \cong E'$$

(over $\bar{K}$) if and only if

$$j(E) = j(E').$$

Proof. See [71, Proposition 1.4 (b)]. □

From the close connection[2] between injections of function fields and isogenies, the only purely inseparable isogenies are those of degree $q = p^e$, where $p = \mathrm{char}(K)$, of the form

$$\pi_q(x, y) = (x^q, y^q).$$

This isogeny (regardless of domain and codomain) is referred to as the ***q*-power Frobenius isogeny**.

There is also close connection between separable isogenies and finite subgroups of E. This following proposition is used a countless number of times throughout this thesis and is essential for working explicitly with isogenies, e.g. in isogeny-based cryptography.

[2] To be precise, the asstociation $E \to K(E)$ and $\varphi \to \varphi^*$ defines a fully faithful functor (see Definition 3.1.5) from the categroy of elliptic curves over K to the category of function-fields over K.

Proposition 2.1.8. *Let E be an elliptic curve.*

(i) *Given a separable isogeny $\varphi : E \to E'$, $\ker\varphi < E$ is a finite subgroup of E satisfying $\#\ker\varphi = \deg\varphi$.*

(ii) *Given a finite subgroup $K < E$, there exists a unique seperable isogeny $\varphi : E \to E''$ (up to isomorphism), such that $\ker\varphi = K$.*

Proof. See [71, Theorem 4.10 (c)] and [71, Proposition 4.12] □

One of the isogenies we have already encountered is denoted $[m]$, the **multiplication-by-m map**. By definition, $\ker[m] = E[m]$, and thus for $\mathrm{char}(K) \nmid m$, combining Proposition 2.1.8 and Proposition 2.1.2 tells us that $[m]$ is an isogeny of degree m^2.

Theorem 2.1.9. *Let $\varphi : E \to E'$ be an isogeny of degree m. Then, there exists a unique isogeny*

$$\widehat{\varphi} : E' \to E$$

such that $\widehat{\varphi} \circ \varphi = [m]$ on E.

Proof. See [71, Theorem 6.1]. □

The isogeny $\widehat{\varphi}$ in the theorem above is referred to as the **dual isogeny** of φ.

Computing isogenies. We now focus on computing isogenies, a central theme in this thesis. The first step to computing large degree isogenies is to "factor" them. We first state this result, for future reference.

Proposition 2.1.10. *Let $\varphi : E_1 \to E_2$ be a separable isogeny, and let $\psi : E_1 \to E_3$ be a (not necessarily separable) isogeny. If $\ker\varphi \subseteq \ker\psi$, then there exists a unique isogeny η making the following diagram commute*

$$\begin{tikzcd}
& E_2 \arrow[dr, dashed, "\exists!\eta"] & \\
E_1 \arrow[ur, "\varphi"] \arrow[rr, "\psi"] & & E_3
\end{tikzcd}$$

Proof. See [71, Corollary III.4.11]. □

Thus, we can always "factor" isogenies, in the following sense.

Corollary 2.1.11. *Let $\varphi : E \to E'$ be an isogeny defined over a field of characteristic p, and write*

$$\deg\varphi = \ell_1\ell_2\cdots\ell_n p^r$$

for (not necessarily distinct) primes ℓ_i distinct from p. Then φ can be written as a composition

$$\varphi = \varphi_1 \circ \varphi_2 \circ \cdots \varphi_n \circ \pi_{p^r},$$

where φ_i are separable isogenies of degree ℓ_i, and π_{p^r} is the p^r-power Frobenius.

Proof. By [71, Corollary II.2.12] we can factor φ as $\varphi' \circ \pi_q$, where φ' is separable of degree $\ell_1 \ell_2 \cdots \ell_n$, which we can further factor into ℓ_i-isogenies by Proposition 2.1.10. □

Thus, we can reduce the problem of computing isogenies of any degree to that of computing separable isogenies of prime degree, for which we can use Proposition 2.1.8. The connection in Proposition 2.1.8 can be made explicit using Vélu's formulae [77]. In essence, computing an isogeny of odd degree ℓ whose kernel is generated by P through Vélu's formulae comes down to computing the **kernel polynomial**

$$h(X) = \prod_{1 \leq m \leq (\ell-1)/2} (X - x([m]P)),$$

as captured by the following proposition.

Proposition 2.1.12. *Let $\varphi : E \to E'$ be a separable isogeny of odd degree ℓ between Weierstrass curves. Then φ can be written in terms of the kernel polynomial $h(X)$ as*

$$\varphi(x, y) = \left(\frac{f(X)}{h(X)^2}, \frac{g(X, Y)}{h(X)^3} \right).$$

for some f, g which depend only on E and h.

Proof. See for instance [47, Section 2.4], which also gives the formulae for f and g, and handles the case $\ell = 2$. □

A straightforward application of Proposition 2.1.12, using product trees, gives an algorithm that runs in $\widetilde{O}(\ell)$ field operations for computing (explicitly, in terms of rational functions) an isogeny of degree n. However, in many cases, what we really want when we are "computing an isogeny" is to have some way of being able to efficiently evaluate it at points. Given a point $P = (x, y)$, we can easily evaluate $\varphi(P)$ in $O(\ell)$ field-operations, again using Proposition 2.1.12 directly, instead of computing the rational functions. This can even be improved to $\widetilde{O}(\sqrt{\ell})$ time using the celebrated $\sqrt{\text{élu}}$-algorithm [5].

Finally, recent advancements based on the SIDH-attacks (See Section 4.3) [8, 55, 65] have made it possible to evaluate ℓ-isogenies in logarithmic time [63]. While it only works under strong conditions[3], it is a huge theoretical breakthrough, that is rapidly finding applications in isogeny-based cryptography [27, 13].

2.2 The Endomorphism Ring

As elliptic curves are abelian groups, the set of isogenies between two elliptic curves, $\mathrm{Hom}(E, E')$, can also be given the structure of an abelian group, by defining addition pointwise, i.e. given $\varphi, \psi \in \mathrm{Hom}(E, E')$, we define

[3] At the very least, it requires knowing the image of a few well-chosen points.

$$(\varphi + \psi)(P) = \varphi(P) + \psi(P), \quad \forall P \in E.$$

The structure of these homsets is given by the following proposition

Proposition 2.2.1. *Let E, E' be two isogenous elliptic curves. Then* $\mathrm{Hom}(E, E')$ *is a free $\mathbb{Z}$-module of rank* $1, 2$ *or* 4.

Proof. See [71, Corollary 7.5]. □

We are especially interested in the case when $E' = E$. In this case, it makes sense to define multiplication by composition. This gives the following definitions.

Definition 2.2.2. Let E be an elliptic curve. The **endomorphism ring** of E is the ring

$$\mathrm{End}(E) := (\mathrm{Hom}(E, E), +, \circ).$$

The **endomorphism algebra** of E is the $\mathbb{Q}$-algebra

$$\mathrm{End}_{\mathbb{Q}}(E) := \mathrm{End}(E) \otimes \mathbb{Q}$$

We first note that the endomorphism algebras are invariant under isogenies.

Proposition 2.2.3. *Let $\varphi : E_1 \to E_2$ be an isogeny. Then* $\mathrm{End}_{\mathbb{Q}}(E_1) = \mathrm{End}_{\mathbb{Q}}(E_2)$

Proof. The isomorphism is given by sending $\alpha \in \mathrm{End}_{\mathbb{Q}}(E_1)$ to $\frac{1}{\deg \varphi}\varphi\alpha\widehat{\varphi} \in \mathrm{End}_{\mathbb{Q}}(E_2)$. □

Second, with our preliminary results on division algebras over $\mathbb{Q}$, determining $\mathrm{End}_{\mathbb{Q}}(E)$ is straightforward.

Proposition 2.2.4. *Let E be an elliptic curve. Then one of the following holds:*

(i) $\mathrm{End}_{\mathbb{Q}}(E) = \mathbb{Q}$,

(ii) $\mathrm{End}_{\mathbb{Q}}(E)$ *is an imaginary quadratic field,*

(iii) $\mathrm{End}_{\mathbb{Q}}(E)$ *is a definite quaternion algebra.*

Proof. Given $\varphi \in \mathrm{End}_{\mathbb{Q}}(E)$, we define

$$\varphi^{-1} = \frac{\widehat{\varphi}}{\deg \varphi} \in \mathrm{End}_{\mathbb{Q}}(E),$$

which is well defined, since $\deg \varphi \in \mathbb{Z}_{>0}$. Then, $\varphi^{-1} \circ \varphi = \varphi \circ \varphi^{-1} = 1$, hence $\mathrm{End}_{\mathbb{Q}}(E)$ is a division $\mathbb{Q}$-algebra. Thus, by verifying that the function

$$\widehat{\ }: \mathrm{End}_{\mathbb{Q}}(E) \to \mathrm{End}_{\mathbb{Q}}(E)$$

given by sending endomorphisms to their dual is a standard involution, Theorem 1.1.3 tells us that $\operatorname{End}_{\mathbb{Q}}(E)$ is either $\mathbb{Q}$, a quadratic number field, or a quaternion algebra. Finally, the corresponding norm

$$\operatorname{nrd}(\varphi) = a\varphi a\widehat{\varphi} = a^2 \deg\varphi, \quad a \in \mathbb{Q}$$

only represent non-negative numbers, which further restricts to imaginary quadratic fields and definite quaternion algebras, thus finishing the proof. □

In fact, the structure of $\operatorname{End}(E)$ **almost** follows directly from this. However, we do not give the full proof.

Theorem 2.2.5. *Let E be an elliptic curve. Then one of the following holds:*

(i) $\operatorname{End}(E) = \mathbb{Z}$*,*

(ii) $\operatorname{End}(E)$ *is an order in an imaginary quadratic field,*

(iii) $\operatorname{End}(E)$ *is a maximal order in* $B_{p,\infty}$*.*

Proof. By Proposition 2.2.1, $\operatorname{End}(E)$ is a finitely generated free $\mathbb{Z}$-module, hence a lattice in $\operatorname{End}_{\mathbb{Q}}(E)$, which proves the theorem whenever $\operatorname{End}_{\mathbb{Q}}$ is commutative. However, in the last case, it is only clear that $\operatorname{End}(E)$ is a definite quaternion order. For a full proof, giving the discriminant of $\operatorname{End}_{\mathbb{Q}}(E)$, and the maximality of $\operatorname{End}(E)$, we refer to the proof in Voight [78, Theorem 42.1.9]. □

As in any ring, the invertible elements $\operatorname{Aut}(E) := \operatorname{End}(E)^\times$ form a group called the automorphism group of E. By the theorem above, this group is very limited.

Corollary 2.2.6. *Let K be a field with* $\operatorname{char}(K) \neq 2, 3$*, let E/K be an elliptic curve, and let* $\operatorname{Aut}(E)$ *denote its automorphism group. Then*

(i) $\# \operatorname{Aut}(E) = 2 \Leftrightarrow j(E) \neq 0, 1728$*,*

(ii) $\# \operatorname{Aut}(E) = 4 \Leftrightarrow j(E) = 1728 \Leftrightarrow \mathbb{Z}[i] \subseteq \operatorname{End}(E), i^2 = -1$*,*

(iii) $\# \operatorname{Aut}(E) = 6 \Leftrightarrow j(E) = 0 \Leftrightarrow \mathbb{Z}[\omega] \subseteq \operatorname{End}(E), \omega^3 = 1, \omega \neq 1$*.*

Proof. All the first implications are given in [71, Theorem III.10.1] (also covering $\operatorname{char}(K) = 2, 3$). The last implication simply comes from looking at $\operatorname{End}(E)^\times$, where $\operatorname{End}(E)$ is as given in Theorem 2.2.5. □

The following is an illustrative example of computing endomorphism rings, in the special case we are interested in, namely elliptic curves over finite fields.

Example 9: Determining an endomorphism ring. Let $p \neq 2, 3, 11$ be a prime, and let

$$E/\mathbb{F}_p : y^2 = x^3 - 1056x + 13552$$

be an elliptic curve. Our first goal is to figure out $\operatorname{End}_{\mathbb{Q}}(E)$. Since E is defined over $\mathbb{F}_p$, it is easy to verify that the p-power Frobenius isogeny π is in fact an endomorphism of E. Further, π is an endomorphism of degree p, hence $\pi \notin \mathbb{Z}$, since the multiplication-by-m map has degree m^2 for all $0 \neq m \in \mathbb{Z}$. Thus $\mathbb{Z} \subsetneq \mathbb{Z}[\pi] \subseteq \operatorname{End}(E)$.

Next, let $\omega \in \mathbb{F}_{p^2}$ satisfy $\omega^2 = -11$, and define

$$\iota((x, y)) = (f_1(x), y f_2(x)),$$

where

$$f_1(x) = \frac{(\omega - 5)x^3 + (-24\omega + 264)x^2 + (-2112\omega - 11616)x + (41536\omega + 147136)}{18(x^2 + (-4\omega - 44)x + (88\omega + 440))},$$
$$f_2(x, y) = \frac{(-\omega - 4)x^3 + (90\omega + 198)x^2 + (264\omega - 6072)x + (-24640\omega + 15488)}{27(x^3 + (-6\omega - 66)x^2 + (264\omega + 1320)x + (-2816\omega - 7744))},$$

which can be verified to be an endomorphism of degree 3. By explicit computation[a], one can further show that

$$([2] \circ \iota - [1])^2 = [-11].$$

We now consider two cases, based on whether -11 is a square in $\mathbb{F}_p$ or not. Assume that -11 is not a square, implying that $\omega \in \mathbb{F}_{p^2} \setminus \mathbb{F}_p$. By quadratic reciprocity, this can be shown to be equivalent with

$$p \mod 11 \in \{2, 6, 7, 8, 10\}.$$

In this case, we have $\omega^p = -\omega$, hence $\pi \circ \iota \neq \iota \circ \pi$, which immediately shows that $\operatorname{End}_{\mathbb{Q}}(E)$ is non-commutative, hence by Proposition 2.2.4, it is a definite quaternion algebra.

As noted, $([2] \circ \iota - [1])^2 = [-11]$, and later we will show that $\pi^2 = [-p]$ (see Proposition 2.3.3) in this case, hence

$$\operatorname{End}_{\mathbb{Q}}(E) = \mathbb{Q} + \iota\mathbb{Q} + \pi\mathbb{Q} + (\iota \circ \pi)\mathbb{Q} \cong \left(\frac{-11, -p}{\mathbb{Q}}\right),$$

where the isomorphism is given by sending $\iota \to \frac{1+\mathbf{i}}{2}$ and $\pi \to \mathbf{j}$.

However, in Example 5, we saw that the order $\mathbb{Z}\langle \iota, \pi \rangle \cong \mathbb{Z}\langle \frac{1+\mathbf{i}}{2}, \mathbf{j} \rangle$ was not maximal, thus we know that $\mathbb{Z}\langle \iota, \pi \rangle \subsetneq \operatorname{End}(E)$.

We can try different maximal orders containing $\mathbb{Z}\langle \iota, \pi \rangle$ as candidates for the endomorphism ring. If the potential endomorphism ring has a basis element of the form $\frac{n\iota + m\pi}{d}$, then this can be confirmed to be an endomorphism of E by checking if $([n]\iota + [m]\pi)(E[d]) = \infty$, since this shows that $[n]\iota + [m]\pi$ can be written as some endomorphism times $[d]$.[b]

Checking this for the maximal order we found in Example 5 indeed confirms that

$$\begin{aligned} \operatorname{End}(E) &= \mathbb{Z} + \iota\mathbb{Z} + (\iota \circ \pi)\mathbb{Z} + \frac{([2] \circ \iota - [1]) \circ ([1] - \pi)}{11}\mathbb{Z} \\ &\cong \mathbb{Z} + \frac{1 + \mathbf{i}}{2}\mathbb{Z} + \frac{\mathbf{j} + \mathbf{k}}{2}\mathbb{Z} + \frac{\mathbf{i} - \mathbf{k}}{11}\mathbb{Z}. \end{aligned}$$

Next, we consider the case when -11 is a square in $\mathbb{F}_p$, i.e. when

$$p \mod 11 \notin \{2, 6, 7, 8, 10\}.$$

In this case $\omega \in \mathbb{F}_p$, and it is easy to see that $(\iota \circ \pi) = (\pi \circ \iota)$. Since $\mathbb{Z}\langle \iota, \pi \rangle \subseteq \operatorname{End}(E)$, and by Theorem 2.2.5, $\operatorname{End}(E)$ is always an imaginary quadratic order or a maximal order in $B_{p,\infty}$, the equality $(\iota \circ \pi) = (\pi \circ \iota)$ implies that ι, π are contained in the same imaginary quadratic order. From the equality $([2] \circ \iota - [1])^2 = [-11]$, we have that $\mathbb{Z}[\iota]$ is the maximal order in $\mathbb{Q}(\sqrt{-11})$, and $\pi = [m] + [n] \circ \iota$ for integers m, n satisfying $m^2 + mn + 3n^2 = p$.

It turns out that there does not exist any other endomorphisms, so in this case, $\operatorname{End}_{\mathbb{Q}}(E) \cong \mathbb{Q}(\sqrt{-1})$, and $\operatorname{End}(E) = \mathbb{Z}[\iota]$.

[a] or, being a bit smarter, one can compute the trace of ι, and derive it from that (and its degree).

[b] This is not particularly effective for large d, however, recent developments have given more advanced techniques, which can indeed perform this "check" in polynomial time in $\log d$ [56, Theorem 4.1].

2.2.1 The Trace of Frobenius

As in Example 9, it can be shown that all elliptic curves defined over $\mathbb{F}_q$ have at least one non-integer endomorphism, e.g. typically the q-power Frobenius endomorphism.[4] This endomorphism is the key to counting the number of points in $E(\mathbb{F}_q)$ efficiently: Clearly,

$$E(\mathbb{F}_q) = \ker(\pi_q - [1]),$$

and further, it can be shown that $\pi_q - [1]$ is always a separable isogeny, hence by Proposition 2.1.8,

$$\# E(\mathbb{F}_q) = \# \ker(\pi_q - [1]) = \deg(\pi_q - [1]).$$

[4] Although, it can actually happen that π_q is an integer. Ironically though, we shall see that these cases imply that the curve has endomorphism ring isomorphic to a maximal order in a quaternion algebra, hence exceptionally many non-integer endomorphisms.

And finally, by using the positive involution in the endomorphism algebra, we can compute

$$\begin{aligned}\deg(\pi_q - [1]) &= \mathrm{nrd}(\pi_q - 1)\\ &= (\pi_q - 1)(\widehat{\pi_q} - 1)\\ &= \pi_q\widehat{\pi_q} - (\pi_q + \widehat{\pi_q}) + 1\\ &= \mathrm{nrd}(\pi_q) - \mathrm{trd}(\pi_q) + 1 = q + 1 - \mathrm{trd}(\pi_q).\end{aligned}$$

which recovers the famous Hasse bound.

Theorem 2.2.7 (Hasse)**.** *Let $E/\mathbb{F}_q$ be an elliptic curve. Then*

$$\#E(\mathbb{F}_q) = q + 1 - t, \quad |t| \leq 2\sqrt{q}.$$

Proof. From the discussion above, we have that $t = \mathrm{trd}(\pi_q)$, which immidiately implies the bound on $|t|$, since $\mathrm{nrd}(\pi_q) = \deg \pi_q = q$. □

From Hasse's theorem, it is clear that computing the number of points on an elliptic curve over a finite field is equivalent to computing the trace of Frobenius.

Example 10: Number of points, and trace of Frobenius. In Example 7, we looked at E over $\mathbb{F}_p$ for $p = 31$, and counted $\#E(\mathbb{F}_p) = 27$ in a rather naïve way. In general, this can be done in polynomial time (in $\log(p)$), by computing the trace of Frobenius using Schoof's algorithm [68, 69]. However, for our particular choice of E, we can do it even more easily. In Example 9, we showed[a] that $\pi \in \mathbb{Z}[\iota]$ (since $p = 31 \equiv 9 \pmod{11}$), where $\iota = \frac{1+\sqrt{-11}}{2}$. Thus, by using Cornacchia's algorithm [21], we can write

$$\pi = [4] + [-3]\iota,$$

and hence read off

$$\#E(\mathbb{F}_p) = p + 1 - \mathrm{t}(\pi) = 31 + 1 - 5 = 27.$$

[a] This is really the same curve since $x^3 - 1056x + 13552 = x^3 + 29x + 5$ in $\mathbb{F}_{31}[x, y]$

As we remark below, the strategy from the previous example generalises well.

Remark 5. *Note that everything about the endomorphism ι from Example 9 remains true even when the curve is defined over $\mathbb{Q}$, when we take $\omega \in \mathbb{Q}(\sqrt{-11})$. Thus this curve clearly has a non-integer endomorphisms over $\overline{\mathbb{Q}}$. Such curves are quite exceptional. In fact, Theorem 2.2.10 from the next section holds over arbitrary fields and implies that up to isomorphism, this is the only curve over $\bar{\mathbb{Q}}$ with CM by the maximal order in $\mathbb{Q}(\sqrt{-11})$.*

We further note that generalising the idea in Example 10 to any ordinary reduction of $E/\bar{\mathbb{Q}}$, where E has complex multiplication leads to a well-known method of constructing elliptic curves where the group of rational points has a given order, typically referred to as the CM-method, which was introduced by Atkin and Morain [2].

2.2.2 Action by the Class Group

We focus briefly on the ordinary case, before moving on to the supersingular case, which is the main topic of this thesis. This will be interesting in its own right, but it also motivates the introduction of orientations later, which allows us to recover a lot of the classical theory on ordinary curves, to the supersingular case.

We start by relating ideals of the endomorphism ring to isogenies. This is a central theme in this thesis. Note that the following definition applies to all cases of *Theorem* 2.2.5, and we will indeed use it actively for both commutative and non-commutative endomorphism rings.

Definition 2.2.8. Let E/K be an elliptic curve with endomorphism ring O, and let $I \subset O$ be an integral O-ideal, with norm coprime to $\mathrm{char}(K)$. We define the ***I*-torsion** to be

$$E[I] := \bigcap_{\alpha \in I} \ker \alpha,$$

and the corresponding **isogeny defined by** I to be

$$\varphi_I : E \to E/E[I].$$

We give a few basic properties of the objects defined above.

Proposition 2.2.9. *Let E/K be an elliptic curve and $\mathbb{Z} \subsetneq O$ be an order, and fix an isomorphism $\iota : O \to \mathrm{End}(E)$. Let $I \subset O$ be an integral O-ideal, with norm coprime to $\mathrm{char}(K)$. Then $\deg \varphi_I = \mathrm{nrd}(I)$. Further, if $I = O\alpha$ is principal, then $\widehat{\varphi_I} = \varphi_{\bar{I}}$.*

Proof. Notice that the second statement follows immediately from the fact that $\varphi_I = \varphi_\alpha = \iota(\alpha)$, and that $\widehat{\iota(\alpha)} = \iota(\overline{\alpha})$.

For the first statement, notice first that for any $\beta \in O$, we have $\iota(\mathrm{nrd}(\beta)) = \iota(\beta\overline{\beta}) = \iota(\beta)\widehat{\iota(\beta)} = \deg \iota(\beta)$, and since ι fixes $\mathbb{Z}$, we conclude that $\mathrm{nrd}(\beta) = \deg \iota(\beta)$. For general ideals, write $I = O\langle N, \gamma \rangle$ (this is always possible, for both imaginary quadratic ideals and quaternion ideals), and so, we can conclude that $\deg \varphi_I = \#(E[N] \cap \ker \iota(\gamma)) = \gcd(N^2, \mathrm{nrd}(\gamma)) = \mathrm{nrd}(I)$. □

The classical class group action is related to the second case of Theorem 2.2.5. Thus, for an imaginary quadratic order $\mathfrak{O}$ and an arbitrary[5] field K, we let $\mathcal{E}\ell\ell_K(\mathfrak{O})$ denote the set of curves over K with endomorphism ring isomorphic to $\mathfrak{O}$. It turns out that given a curve $E \in \mathcal{E}\ell\ell_K(\mathfrak{O})$ and an invertible ideal $\mathfrak{a} \subset \mathfrak{O}$, the curve $E_\mathfrak{a} := E/E[\mathfrak{a}]$ also satisfies $E_\mathfrak{a} \in \mathcal{E}\ell\ell_K(\mathfrak{O})$. Further, the following theorem tells us that this induces a free and transitive action by $\mathrm{Cl}(\mathfrak{O})$ on $\mathcal{E}\ell\ell_K(\mathfrak{O})$.

[5] i.e. K is typically not the field of fractions of $\mathfrak{O}$

Theorem 2.2.10. *Let $\mathfrak{O}$ be an imaginary quadratic order and K a field. Then,* $\mathrm{Cl}(\mathfrak{O})$ *acts freely and transitively on the set $\mathcal{E\ell\ell}_K(\mathfrak{O})$, where the group action is defined by*

$$\begin{aligned} \mathrm{Cl}(\mathfrak{O}) \times \mathcal{E\ell\ell}_K(\mathfrak{O}) &\to \mathcal{E\ell\ell}_K(\mathfrak{O}), \\ [\mathfrak{a}] \star E &= E_{\mathfrak{a}}. \end{aligned}$$

Proof. See the commutative case in [79, Theorem 4.5]. □

As we will see in Chapter 3, in the supersingular case every isogeny can be realised as an isogeny defined by a (quaternion) ideal. However, this is not true in the ordinary case. We define three types of isogenies:

Definition 2.2.11. Let $\varphi : E_1 \to E_2$ be an isogeny defined over K, between curves $E_1/K, E_2/K$ satisfying $\mathrm{End}(E_1) \cong \mathfrak{O}_1$, and $\mathrm{End}(E_2) \cong \mathfrak{O}_2$ for imaginary quadratic orders $\mathfrak{O}_1, \mathfrak{O}_2$. Then φ is called

(i) **horizontal** whenever $\mathfrak{O}_1 = \mathfrak{O}_2$,

(ii) **ascending** whenever $\mathfrak{O}_1 \subsetneq \mathfrak{O}_2$,

(iii) **descending** whenever $\mathfrak{O}_1 \supsetneq \mathfrak{O}_2$.

With these definitions, it can be shown that the isogenies defined by quadratic ideals are always ascending or horizontal.

Notice that one of the conditions above is always satisfied, because of Proposition 2.2.3. In fact, for descending isogenies, we always have $[\mathfrak{O}_1 : \mathfrak{O}_2] = \deg \varphi$, and similar for ascending isogenies.

Example 11: Isogeny volcano. We show how arranging ℓ-isogenies suitably, based on whether they are horizontal, ascending or descending, leads to the beautiful **isogeny-volcanoes**. See Figure 2.1 for a visual accompaniment to this example.

Let $\mathfrak{O}$ be an order with $h(\mathfrak{O}) = 6$, and let $\mathfrak{l} \subset \mathfrak{O}$ be an ideal with $\mathrm{n}(\mathfrak{l}) = 2$ such that $[\mathfrak{l}] \in \mathrm{Cl}(\mathfrak{O})$ is a generator of the whole class group, and consider a curve $E/\mathbb{F}_q \in \mathcal{E\ell\ell}_K(\mathfrak{O})$. Taking the isogeny corresponding to $\mathfrak{l}$, we end up on another curve $E'/\mathbb{F}_q \in \mathcal{E\ell\ell}_K(\mathfrak{O})$ by Theorem 2.2.10, and we can repeat the procedure from E'. Doing this 6 times corresponds to the action of $[\mathfrak{l}]^6$, which is trivial by assumption, hence we end up back at E. This cycle is commonly referred to as the **crater** of the volcano.

Notice that we could have done the same procedure with $\overline{\mathfrak{l}}$ instead, corresponding to a different 2-isogeny. However, by Proposition 2.1.2, there exist 3 subgroups of order 2 on E, hence there exists a third isogeny of degree 2, not corresponding to an $\mathfrak{O}$-ideal. This isogeny is thus descending, and hence we land on a curve E'' with $\mathrm{End}(E'') \cong \mathbb{Z} + 2\mathfrak{O}$. The order $\mathbb{Z} + 2\mathfrak{O}$ has no invertible ideals of norm 2, hence all 2-isogenies from E'' must either be ascending or descending, and one can show that the only ascending is the dual of the isogeny we took to land on E''.

We can keep descending from E'' as long as our curves have full 2-torsion over $\mathbb{F}_q$.

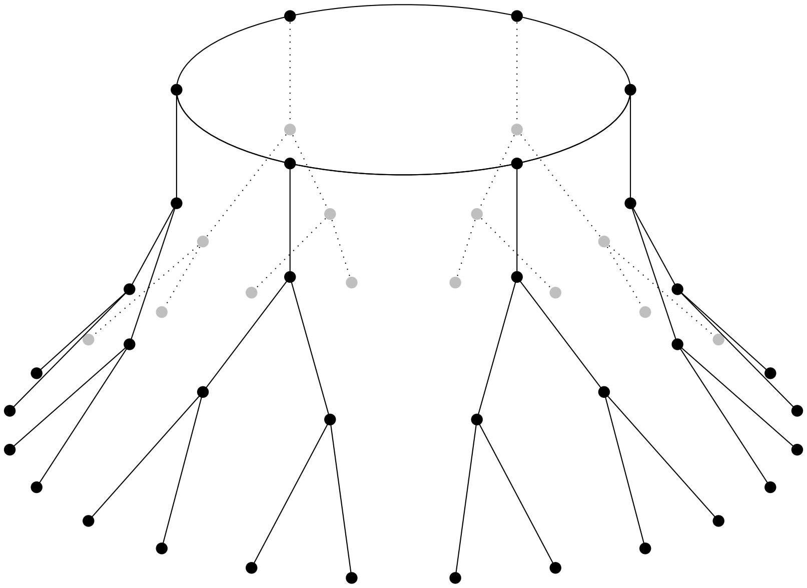


Figure 2.1: An ordinary 2-isogeny volcano. Nodes correspond to elliptic curves over $\mathbb{F}_q$, and edges correspond to $\mathbb{F}_q$-rational 2-isogenies.

> However, this cannot be repeated indefinitely; We eventually end up at a curve F with $\mathrm{End}(F) \cong \mathbb{Z} + 2^d\mathfrak{O}$. Since F is defined over $\mathbb{F}_q$, we know a non-integer endomorphism of order q, namely π_q, the q-power Frobenius isogeny (the non-triviality of this isogeny can, for instance, be seen from the trace, together with the fact that F is ordinary). We thus know that $\pi_q = [n] + 2^d\delta[m]$, for some generator δ of $\mathrm{End}(F)$. Simply by looking at the norm, we can thus bound $d \leq \log_2(q)/2$, and we know that F does not have full 2-torsion over $\mathbb{F}_q$. Curves at this level of the volcano are referred to as the **floor** of the volcano.

2.3 Supersingularity

It turns out that over $\overline{\mathbb{F}}_p$, all but a finite number of (isomorphism classes) of elliptic curves have $\mathrm{End}(E)$ isomorphic to an imaginary quadratic order. We will however be mainly focused on the special case.

Definition 2.3.1. Let $E/\mathbb{F}_p$ be an elliptic curve. Then E is said to be **supersingular** if $\mathrm{End}(E)$ is isomorphic to a maximal order in a quaternion algebra ramified at p and ∞.

Next, we will list some key properties of supersingular elliptic curves, before generalising the class group action from Section 2.2.2 to these types of curves.

2.3.1 Properties of Supersingular Curves

The definition of supersingularity we use is only one of many equivalent characterisations. We list a few that are especially important to us.

Theorem 2.3.2. *Let $E/\mathbb{F}_q$ be an elliptic curve, where $q = p^r$. Let $\pi : E \to E^{(p)}$ denote the p-power Frobenius isogeny, and let π_q denote the q-power Frobenius endomorphism.*

Then E is supersingular $\Leftrightarrow$ $[-q] = \psi \circ \pi_q^2$ for some automorphism $\psi \in \operatorname{End}(E)$ $\Leftrightarrow$ there exists $E' \cong E$ such that E' is defined over $\mathbb{F}_{p^2}$, and $\pi^2 = [-p]$ on E'.

Proof. This all follows readily from [71, Theorem V.3.1], which lists more equivalent definitions. □

Another equivalent characterisation that is typically stated is that $E/\mathbb{F}_q$ is supersingular if and only if $\operatorname{trd}(\pi_q) \equiv 0 \pmod p$. We give only one direction of this in the proposition below, but with a more accurate statement, which is due to Waterhouse [79, Theorem 4.1].

Proposition 2.3.3. *Let $p > 3$ be a prime, $q = p^r$, let $E/\mathbb{F}_q$ be a supersingular elliptic curve, and let π_q denote the q-power Frobenius on E.*

(i) *If $r \equiv 1 \pmod 2$, then $\pi_q^2 = [-q]$, and $\operatorname{trd}(\pi_q) = 0$.*

(ii) *If $r \equiv 0 \pmod 2$, then*

$$\operatorname{trd}(\pi_q) \in \begin{cases} \{\pm 2\sqrt{q}\} & \text{if } j(E) \neq 0, 1728 \\ \{0, \pm 2\sqrt{q}\} & \text{if } j(E) = 1728 \\ \{\pm\sqrt{q}, \pm 2\sqrt{q}\} & \text{if } j(E) = 0 \end{cases}$$

Proof. From Theorem 2.3.2, we know that

$$\pi_q^2 = \varphi \circ [q]$$

for some automorphism φ. We first show the case $r \equiv 1 \pmod 2$. As an abstract element in $\operatorname{End}(E)$,

$$\pi_q = \sqrt{\omega q},$$

for some $\omega \in \operatorname{Aut}(E)$. Corollary 2.2.6 implies that there are only a few possiblities for $\operatorname{Aut}(E)$. It is clear that the only choices of ω in the set of possible automorphisms that give elements that are quadratic over $\mathbb{Q}$ are $\omega = \pm 1$ (here we use that $\sqrt{q} \notin \mathbb{Q}$). The case $\omega = 1$ is excluded, because $\sqrt{q} \in \mathbb{R} \setminus \mathbb{Q}$, hence not a possible element of $\operatorname{End}(E)$. Thus $\pi_q^2 = [-q]$ and $\operatorname{trd}(\pi_q) = 0$.

Next, we show the case $r \equiv 0 \pmod 2$. Keep in mind that here, $\sqrt{q} \in \mathbb{Z}$. We go through every case:

$\omega = 1$**:** In this case, $\pi_q = \pm\sqrt{q}$, in which case $\operatorname{trd}(\pi_q) = \pm 2\sqrt{q}$.

$\omega = -1$**:** In this case, we get that $\pi_q = \pm\sqrt{-q} = \pm\iota\sqrt{q}$ for some unit ι, which must satisfy $\iota^2 = -1$. This implies that $j(E) = 1728$, and $\operatorname{trd}(\pi_q) = 0$.

$\omega^3 = 1, \omega \neq 1$**:** In this case $\pi_q = \pm\sqrt{\omega q} = \pm\zeta\sqrt{q}$ for some unit ζ, which must satisfy $\zeta^2 = \omega$, and hence $\zeta^6 = 1$. This implies $j(E) = 0$ and $\operatorname{trd}(\pi_q) = \pm\sqrt{q}$.

All further choices of $\omega \in \operatorname{Aut}(E)$ clearly give elements which are not quadratic over $\mathbb{Q}$, which is impossible. □

Another consequence of the relationship between supersingularity and the (possible) triviality of the Frobenius endomorphism is the following.

Corollary 2.3.4. *Let E be a supersingular elliptic curve, and let $\varphi : E \to E'$ be an isogeny. Then E' is also supersingular, and furthermore, φ can be defined over $\mathbb{F}_{p^2}$, by pre- and post-composing with appropriate isomorphisms.*

Proof. First, by Proposition 2.2.3, the endomorphism algebras are invariant under isogenies, hence E' is also supersingular. Second, since the Frobenius is defined over the base field, we only need to consider the case when φ is separable. By Theorem 2.3.2, we can assume that E is defined over $\mathbb{F}_{p^2}$ and satisfies $\pi_E^2 = [-p]$. Thus, every subgroup of E, and in particular $\ker\varphi$, is closed under the action of Galois (which is generated by the p^2-Frobenius), and by Proposition 2.1.8, φ can be defined over $\mathbb{F}_{p^2}$, by post-composing with an appropriate isomorphism. □

Thus, we see that when working with supersingular isogeny-graphs, it is always sufficient to consider curves and isogenies defined over $\mathbb{F}_{p^2}$. In light of Proposition 2.3.3, the isomorphism class of supersingular elliptic curves which satisfy $\pi^2 = [-p]$ are somewhat special. These curves are of course always defined over $\mathbb{F}_{p^2}$, and moreover, they are isogenous to curves defined over $\mathbb{F}_p$. When restricting to these curves, we can be even more accurate with the number of rational points, and the group structure. See for instance Paper 1, Theorem 1.

Still, the fact that we need only work with *curves* and *isogenies* defined over $\mathbb{F}_{p^2}$ does not imply that we never work with *points* over larger field extensions. In fact, from Corollary 2.3.4 we know that the isogeny generated by any point P can be made $\mathbb{F}_{p^2}$-rational, even if the point is over a larger field extension. This simple point is one of the main themes in Paper 1 and Paper 3.

Example 12: Rational isogeny from irrational point. Let $p = 109$, and fix $\mathbb{F}_{p^2} := \mathbb{F}_p(\omega), \omega^2 = -11$, and look at the supersingular elliptic curve

$$E : y^2 = x^3 + (31\omega + 79)x + (39\omega + 86).$$

Let $\mathbb{F}_{p^4} = \mathbb{F}_p(\zeta), \zeta^4 + 11\zeta^2 + 98\zeta + 6 = 0$. It can be verified that $\mathbb{F}_{p^2} \hookrightarrow \mathbb{F}_{p^4}$ by linearly extending the map sending ω to $43\zeta^3 + 40\zeta^2 + 70\zeta + 56$. Take the point

$$P = (6\zeta^3 + 36\zeta^2 + 63\zeta + 65, 24\zeta^3 + 36\zeta^2 + 54\zeta),$$

which can be confirmed to have order 6. Even though $P \notin E(\mathbb{F}_{p^2})$, we can compute its corresponding isogeny φ, which has kernel polynomial

$$\begin{aligned} h(X) &= (X - P)(X - [2]P)(X - [3]P) \\ &= X^3 + (38\omega + 15)X^2 + (62\omega + 62)X + (23\omega + 61) \in \mathbb{F}_{p^2}[X], \end{aligned}$$

thus it is clear that φ is also defined over $\mathbb{F}_{p^2}$.

Interestingly, by composing with an appropriate isomorphism, φ can actually be made into an endomorphism of E. This will come back later.

2.3.2 Orientations and Action by the Class Group

Before studying all isogenies between supersingular elliptic curves in general (which is the focus of Chapter 3), we first give the curves some extra structure and show how we can neatly recover a similar theory to that found in Section 2.2.2. The extra structure is familiar from Section 1.4.

Definition 2.3.5. Let $\mathfrak{O} \subseteq K$ be an imaginary quadratic order in K, and let E be a supersingular elliptic curve. A **K-orientation** on E is an embedding

$$\iota : K \hookrightarrow \mathrm{End}_{\mathbb{Q}}(E).$$

Further, if ι induces an optimal embedding (Definition 1.4.1) of $\mathfrak{O}$ into $\mathrm{End}(E)$, we also refer to $\iota\mid_{\mathfrak{O}}$ as a **primitive $\mathfrak{O}$-orientation**. Finally, we refer to the pair (E, ι) as a K-oriented curve, and the pair $(E, \iota\mid_{\mathfrak{O}})$ as a primitively $\mathfrak{O}$-oriented curve.

Given a K-oriented curve (E_1, ι_1), an isogeny $\varphi : E_1 \to E_2$ induces a K-orientation on E_2 as

$$\begin{aligned} \varphi_*(\iota_1) &: K \to \mathrm{End}_{\mathbb{Q}}(E_2), \\ \varphi_*(\iota_1)(\alpha) &= \frac{1}{\deg \varphi} \varphi \iota_1(\alpha) \widehat{\varphi}. \end{aligned}$$

Correspondingly, we want isogenies between K-oriented elliptic curves to respect the orientation. This motivates the following definition.

Definition 2.3.6. Let $(E_1, \iota_1), (E_2, \iota_2)$ be K-oriented elliptic curves. A K-oriented isogeny

$$\varphi : (E_1, \iota_1) \to (E_2, \iota_2)$$

is an isogeny $\varphi : E_1 \to E_2$ such that $\iota_2 = \varphi_*(\iota_1)$.

Isomorphisms between primitively K-oriented curves are precisely K-oriented isogenies with an inverse. We denote by $SS^{\mathrm{pr}}_{\mathfrak{O}}(p)$ the set of primitively $\mathfrak{O}$-oriented elliptic curves over $\overline{\mathbb{F}}_p$ up to K-oriented isomorphism.

Remark 6. *Let $K = \overline{\mathbb{F}}_p$. Note the similarity (and differences!) between $\mathcal{E}\ell\ell_K(\mathfrak{O})$ and $SS^{\mathrm{pr}}_{\mathfrak{O}}(p)$: The former refers to elliptic curves with endomorphism ring isomorphic to $\mathfrak{O}$, and the latter refers to elliptic curves with an optimal embedding of $\mathfrak{O}$ into their endomorphism ring. The former are by definition ordinary curves, and the latter are, by definition, supersingular.*

Given $(E, \iota) \in SS^{\mathrm{pr}}_{\mathfrak{O}}(p)$ and an ideal $\mathfrak{a} \subset \mathfrak{O}$, by slight abuse of notation, we denote by $\varphi_{\mathfrak{a}}$ the isogeny generated by the ideal $\mathrm{End}(E)\iota(\mathfrak{a})$ (see Definition 2.2.8), and $E_{\mathfrak{a}} := \varphi_{\mathfrak{a}}(E)$. When $\mathfrak{a}$ is an invertible ideal, it can again be shown that $\iota_{\mathfrak{a}} := \varphi_{\mathfrak{a}*}(\iota)$ is a primitive $\mathfrak{O}$-orientation on $E_{\mathfrak{a}}$, such that $(E_{\mathfrak{a}}, \iota_{\mathfrak{a}}) \in SS^{\mathrm{pr}}_{\mathfrak{O}}(p)$.

We can now state an analogue of Theorem 2.2.10, due to Onuki.

Theorem 2.3.7. *Let $\mathfrak{O} \subset K$ be an imaginary quadratic order, and let p be a non-split prime in K. Assume further that the p does not divide the conductor of $\mathfrak{O}$. Then, $\mathrm{Cl}(\mathfrak{O})$ acts freely on the set $SS^{\mathrm{pr}}_{\mathfrak{O}}(p)$ in $1 - (\frac{d_K}{p})$ orbits, where the group action is defined by*

$$\begin{aligned} \mathrm{Cl}(\mathfrak{O}) \times SS^{\mathrm{pr}}_{\mathfrak{O}}(p) &\to SS^{\mathrm{pr}}_{\mathfrak{O}}(p), \\ [\mathfrak{a}] \star (E, \iota) &= (E_{\mathfrak{a}}, \iota_{\mathfrak{a}}). \end{aligned}$$

Proof. Onuki proves this by looking at supersingular reductions of curves over number fields with complex multiplication (i.e. using Theorem 2.2.10 over a suitable number field) [57]. We will later give a completely different proof of the statement, using optimal embeddings in quaternion orders, and the Deuring correspondence. □

Further, for K-oriented isogenies, we get analogues of Definition 2.2.11. Let $\varphi : (E_1, \iota_1) \to (E_2, \iota_2)$ be a K-oriented isogeny, and let $\mathfrak{O}_1 = \iota_1^{-1}(\mathrm{End}(E_1)), \mathfrak{O}_2 = \iota_2^{-1}(\mathrm{End}(E_2))$, i.e. (E_i, ι_i) are primitively $\mathfrak{O}_i$-oriented curves. Then φ is again referred to as either horizontal, ascending, or descending depending on whether $\mathfrak{O}_1 = \mathfrak{O}_2$, $\mathfrak{O}_1 \subsetneq \mathfrak{O}_2$, or $\mathfrak{O}_1 \supsetneq \mathfrak{O}_2$. Thus, we can again draw a (this time, infinite!) volcano.

Example 13: Oriented isogeny volcano. Let $p = 109$, and fix $\mathbb{F}_{p^2} := \mathbb{F}_p(\omega), \omega^2 = -11$. In Example 12, we saw that the elliptic curve

$$E : y^2 = x^3 + (31\omega + 79)x + (39\omega + 86)$$

was supersingular, and further, that it had an endomorphism φ of degree 6 generated by

$$K = (6\zeta^3 + 36\zeta^2 + 63\zeta + 65, -) \in E(\mathbb{F}_{p^4}),$$

where $\mathbb{F}_{p^4} = \mathbb{F}_p(\zeta), \zeta^4 + 11\zeta^2 + 98\zeta + 6 = 0.$

It can be shown that φ has trace 1, and thus it defines an embedding

$$\begin{gathered}\iota : \mathbb{Q}(\delta) \hookrightarrow \operatorname{End}_{\mathbb{Q}}(E),\\ \iota(\delta) = \varphi, \quad \delta = \frac{1+\sqrt{-23}}{2}.\end{gathered}$$

This embedding gives a (necessarily) primitive $\mathbb{Z}[\delta]$-orientation on E. As we saw in Example 3, $\mathrm{Cl}(\mathbb{Z}[\delta])$ has order 3, and is for instance generated by the ideal

$$\mathfrak{l} = \mathbb{Z}3 \oplus \mathbb{Z}\frac{1-\sqrt{-23}}{2}.$$

Consider now the four isogenies of degree 3 from E. Two of them are equal to $\varphi_{\mathfrak{l}}$ and $\varphi_{\overline{\mathfrak{l}}}$, and thus their co-domains are again primitively $\mathbb{Z}[\delta]$-oriented by Theorem 2.3.7, while the last two 3-isogenies are descending. Similarly to Example 11, we can continue like this, creating, this time, an infinite volcano!

However, note that we could have started with a *different* orientation on E by defining it as

$$\begin{gathered}\iota : \mathbb{Q}(\delta) \hookrightarrow \operatorname{End}_{\mathbb{Q}}(E),\\ \iota(\delta) = \widehat{\varphi}.\end{gathered}$$

To see that this is truly a different oriented curve, note that E only has the automorphisms $[\pm 1]$, neither of which takes φ to $\widehat{\varphi}$.

Taking $(E, \widehat{\varphi})$ as our starting oriented curve instead, gives a *new* volcano, that is not connected to the first one we made. The fact that there are two vulcanoes correspond to the fact that there are two orbits by Theorem 2.3.7. See Figure 2.2 for an illustration.

2.4 Level Structures

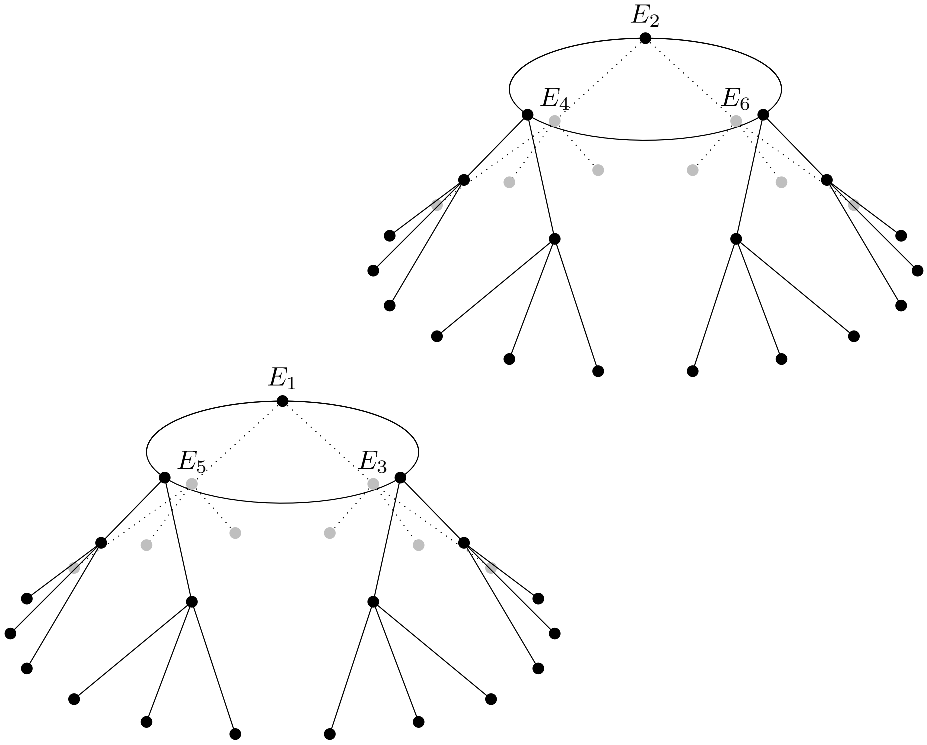
We take a short interlude on Γ-level structures, before continuing on to the Deuring correspondence. In the context of this thesis, the material in this subsection is only relevant for Paper 5.

From Proposition 2.1.2, we know that for any curve E/K, and $N \nmid \mathrm{char}(K)$,

$$E[N] \cong \mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z}.$$

The isomorphism is however highly non-canonical: A choice of isomorphism Φ essentially comes down to choice of torsion-basis $E[N] = \langle P, Q \rangle$, i.e.

$$\begin{gathered}\Phi : \mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z} \to E[N],\\ \Phi((1,0)) = P,\end{gathered}$$



E_i	$\ker\varphi$
$E_1 : y^2 = x^3 + (31\omega + 79)x + (39\omega + 86)$	$\langle(6\zeta^3 + 36\zeta^2 + 63\zeta + 65, -)\rangle$
$E_2 : y^2 = x^3 + (31\omega + 79)x + (39\omega + 86)$	$\langle(57, -)\rangle$
$E_3 : y^2 = x^3 + (14\omega + 23)x + (3\omega + 15)$	$\langle(28\zeta^3 + 59\zeta^2 + 76\zeta + 2, -)\rangle$
$E_4 : y^2 = x^3 + (14\omega + 23)x + (3\omega + 15)$	$\langle(63\zeta^3 + 51\zeta^2 + 62\zeta + 107, -)\rangle$
$E_5 : y^2 = x^3 + (95\omega + 23)x + (106\omega + 15)$	$\langle(46\zeta^3 + 58\zeta^2 + 47\zeta + 95, -)\rangle$
$E_6 : y^2 = x^3 + (95\omega + 23)x + (106\omega + 15)$	$\langle(81\zeta^3 + 50\zeta^2 + 33\zeta + 56, -)\rangle$

Figure 2.2: An oriented 3-isogeny volcano. Even though the ideals above 3 generate the whole class group, there still exist two volcanos. The $\mathfrak{O}$-oriented curves on the top are marked, with corresponding generators of the orientations available in the table. In the table, $\omega^2 + 11 = 0$ and $\zeta^4 + 11\zeta^2 + 98\zeta + 6 = 0$. The volcanos are not drawn past level 2, though there are really infinite levels.

$$\Phi((0,1)) = Q.$$

Informally, a Γ-level structure is essentially a choice of basis, up to scaling by elements, where the elements we are allowed to scale by depend on Γ.

Definition 2.4.1. Let E/K and N be as above, and let $\Gamma < \mathbf{GL}_2(\mathbb{Z}/N\mathbb{Z})$ be a subgroup. A **Γ-level structure** on E is a choice of isomorphism

$$\Phi : \mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z} \to E[N],$$

up to pre-composition by an element $H \in \Gamma$, i.e. $\Phi' \sim \Phi$ if $\Phi' = \Phi \circ H$. The tuple (E, Φ) is referred to as an **elliptic curve with Γ-level structure**.

Some popular choices of subgroups Γ have more natural interpretations. As a first trivial example, consider

$$\Gamma = \mathbf{GL}_2(\mathbb{Z}/N\mathbb{Z}).$$

This is referred to as the **trivial level structure**, since any two choices of isomorphism Φ differ by a change of basis, and thus each elliptic curve has exactly one choice of such a level structure. Similarly, the second trivial example is the **full N-level structure**, given by

$$\Gamma_{\mathrm{id}} = \left\{ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right\},$$

where each isomorphism Φ (i.e. each choice of basis of $E[N]$) corresponds to a unique level structure.

A few non-trivial examples include the **Borel level structure** given by

$$\Gamma_N = \left\{ \begin{pmatrix} a & b \\ 0 & c \end{pmatrix} \middle| \, a, b, c \in \mathbb{Z}/N\mathbb{Z}, ac \in (\mathbb{Z}/N\mathbb{Z})^\times \right\},$$

which correspond to a choice of subgroup of order N. To see this, assume that $\langle P \rangle$ and $\langle P' \rangle$ are two subgroups of E. The generators can be extended to a choice of basis $E[N] = \langle P, Q \rangle$ and $E[N] = \langle P', Q' \rangle$, respectively. Then, P and P' generate the same subgroup if and only if

$$P' = [a]P, \quad \text{and} \quad Q' = [b]P + [c]Q,$$

i.e. their level structures differ by an element of Γ. Similar examples are given by

$$\Gamma_N^1 = \left\{ \begin{pmatrix} 1 & b \\ 0 & c \end{pmatrix} \middle| \, b, c \in \mathbb{Z}/N\mathbb{Z}, c \in (\mathbb{Z}/N\mathbb{Z})^\times \right\},$$

and

$$\Gamma_N^{0,0} = \left\{ \begin{pmatrix} a & 0 \\ 0 & c \end{pmatrix} \middle| \, a, c \in \mathbb{Z}/N\mathbb{Z}, ac \in (\mathbb{Z}/N\mathbb{Z})^\times \right\},$$

which correspond to a point of order N, and two subgroups of order N respectively.

Isogenies between curves with level structure. When working with curves with level structure, we want the isogenies between them to respect the level structures.

Definition 2.4.2. Let (E, Φ) and (E', Φ') we two elliptic curves with Γ-level structure. An isogeny (which respects the level structure)

$$\varphi : (E, \Phi) \to (E', \Phi')$$

is a usual isogeny $\varphi : E \to E'$ of elliptic curves, which satisfies $\Phi' \sim \varphi \circ \Phi$, i.e. if Φ and Φ' are determined by the bases P, Q and P', Q' respectively, then

$$M \begin{pmatrix} \varphi(P) \\ \varphi(Q) \end{pmatrix} = \begin{pmatrix} P' \\ Q' \end{pmatrix}$$

for some $M \in \Gamma$.

Similarly, an isomorphism that respects the level structure is an isogeny

$$\varphi : (E, \Phi) \to (E', \Phi')$$

of degree 1. However, there is one tricky thing to point out: When studying elliptic curves with Γ-level structure up to isomorphism, we may or may not mean up to isomorphism *that respects the level structure.* If we allow all isomorphisms, then this means that given an automorphism $\alpha \in \mathrm{Aut}(E)$, (E, Φ) and $(E, \alpha \circ \Phi)$ are identified, even if they do not differ by an element in Γ, e.g. if $\alpha = [-1]$, they are identified even if

$$\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \notin \Gamma.$$

See [17, Remark 1.3] and Remark 3.7 in Paper 5 for two examples of this annoyance.

Level structures for CM curves. Next, let $\mathfrak{O}$ be an imaginary, quadratic order. In Paper 5, we shall see that when $\mathfrak{O} = \mathrm{End}(E)$ (or more generally, when $\mathfrak{O}$ primitively embeds in $\mathrm{End}(E)$, as will be discussed in Section 2.3.2), it makes sense to consider another type of level structure. This observation is based on the following lemma.

Lemma 2.4.3. *Let $\mathfrak{O}$ be an imaginary quadratic order, and let E be an elliptic curve, such that $\mathfrak{O}$ injects into* $\mathrm{End}(E)$ *Then $E[\mathfrak{a}]$ has the structure of an $\mathfrak{O}$-module, where multiplication by $\alpha \in \mathfrak{O}$ is given by applying the corresponding endomorphism of E.*

Further, given any ideal $\mathfrak{m} \subset \mathfrak{O}$ such that $\#E[\mathfrak{m}] = \mathrm{n}(\mathfrak{m})$, we have

$$E[\mathfrak{m}] \cong \mathfrak{O}/\mathfrak{m}$$

as $\mathfrak{O}$-modules.

Proof. See Paper 5, Lemma 3.1. We note that for principal ideals, this is a well-known result due to Lenstra [53]. □

This allows us to consider more general[6] Γ-level structures, where $\Gamma < \mathbf{GL}(\mathfrak{O}/\mathfrak{m})$, for

[6] Note that this is indeed a generalisation, as $\mathfrak{O}/N\mathfrak{O} \cong \mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z}$.

any ideal $\mathfrak{m} \subset \mathfrak{O}$. In Paper 5 we show a that the class group action of Theorem 2.2.10 (or, correspondingly, Theorem 2.3.7 when $\mathfrak{O} \subsetneq \mathrm{End}(E)$) indeed generalises to so-called **generalised class groups**: these groups act freely and transitively on curves with appropriate Γ-level structure.

Chapter 3

The Deuring Correspondence

In this section, we connect the material from Chapter 1 and Chapter 2, through the focal concept in this thesis: the Deuring correspondence. Deuring was the first to prove that there is a bijection between maximal orders in $B_{p,\infty}$, and Galois conjugacy classes of supersingular j-invariants in $\overline{\mathbb{F}}_p$ [36], though the equivalence of categories is due to Kohel [47].

3.1 Just Enough Category Theory

Our goal is to state the Deuring correspondence as an equivalence of categories, hence we restate the main definition here, for readers not familiar with category theory. For a gentle but thorough introduction, see Leinster [51].

Definition 3.1.1. A (locally small) **category** $\mathcal{A}$ is a collection of objects $\mathrm{Ob}\mathcal{A}$, together with sets $\mathrm{Hom}(A, B)$ for each $A, B \in \mathrm{Ob}\mathcal{A}$, and a composition function

$$\circ : \mathrm{Hom}(B, C) \times \mathrm{Hom}(A, B) \to \mathrm{Hom}(A, C)$$

satisfying the properties:

(i) For all $f \in \mathrm{Hom}(A, B), g \in \mathrm{Hom}(B, C), h \in \mathrm{Hom}(C, D)$, we have

$$h \circ (g \circ f) = (h \circ g) \circ f.$$

(ii) For all $A \in \mathrm{Ob}\mathcal{A}$, we have an **identity** $1_A \in \mathrm{Hom}(A, A)$ satisfying

$$f \circ 1_A = f, \quad 1_A \circ g = g,$$

for all $f \in \mathrm{Hom}(A, B)$ and $g \in \mathrm{Hom}(C, A)$.

Note that we will typically write $A \in \mathcal{A}$ when we mean $A \in \mathrm{Ob}\mathcal{A}$.

One of the main categories of interest is given in the following example, which can easily be verified to satisfy the properties of a category.

> **Example 14: The category of supersingular elliptic curves.** Fix a prime p, and let $\mathcal{SS}_p$ denote the category whose objects are supersingular elliptic curves over $\overline{\mathbb{F}}_p$, and $\mathrm{Hom}(E, E')$ are the sets of isogenies $\varphi : E \to E'$. Setting the identity morphism to be $[1] \in \mathrm{Hom}(E, E) = \mathrm{End}(E)$, it is immediately obvious that this satisfies all the axioms of a category.

Next, we define maps between categories.

Definition 3.1.2. Let $\mathcal{A}, \mathcal{B}$ be categories. A **covariant functor** $F : \mathcal{A} \to \mathcal{B}$ is a function

$$F : \mathrm{Ob}\mathcal{A} \to \mathrm{Ob}\mathcal{B}$$

such that for all $A, B \in \mathcal{A}$, there is a function

$$F : \mathrm{Hom}(A, B) \to \mathrm{Hom}(F(A), F(B))$$

satisfying $F(g \circ f) = F(g) \circ F(f)$ and $F(1_A) = 1_{F(A)}$.

A **contravariant** functor is similar, except that for all $A, B \in \mathcal{A}$, there is instead a function

$$F : \mathrm{Hom}(A, B) \to \mathrm{Hom}(F(B), F(A))$$

such that $F(g \circ f) = F(f) \circ F(g)$.

We can also take this further, and consider maps between functors.

Definition 3.1.3. Let $F, G : \mathcal{A} \to \mathcal{B}$ be two functors. A **natural transformation** η is a family of maps in $\mathcal{B}$ parameterised by the objects in $\mathcal{A}$, such that for each morphism $f : A \to B$ in $\mathcal{A}$ the square

$$\begin{tikzcd}
F(A) \arrow[r, "F(f)"] \arrow[d, "\eta_A"'] & F(B) \arrow[d, "\eta_B"] \\
G(A) \arrow[r, "G(f)"'] & G(B)
\end{tikzcd}$$

commutes. Further, η is called a **natural isomorphism** if all the η_A are isomorphisms in $\mathcal{B}$ for all $A \in \mathcal{A}$.

Next, we wish to have some way of describing when two categories are the same. The obvious definition of isomorphism between two categories (i.e. functors with an inverse) turns out to be an unreasonably strict definition. However, if we allow the functors to be "almost" inverses (or **quasi-inverses**), we get the following useful definition.

Definition 3.1.4. Let $\mathcal{A}, \mathcal{B}$ be categories, and let $F : \mathcal{A} \to \mathcal{B}$ be a functor. Then F is said to be an **equivalence of categories** if there exists a functor $G : \mathcal{B} \to \mathcal{A}$, and natural isomorphisms

$$\eta : 1_{\mathcal{A}} \to G \circ F, \quad \epsilon : F \circ G \to 1_{\mathcal{B}}.$$

Finally, instead of explicitly constructing the quasi-inverse, there is an alternate, and very useful criterion, for determining when a functor is an equivalence. For this, we need the following definitions

Definition 3.1.5. A functor $F : \mathcal{A} \to \mathcal{B}$ is said to be

- **Full** if $F : \mathrm{Hom}(A, B) \to \mathrm{Hom}(F(B), F(A))$ is surjective for all $A, B \in \mathcal{A}$.
- **Faithful** if $F : \mathrm{Hom}(A, B) \to \mathrm{Hom}(F(B), F(A))$ is injective for all $A, B \in \mathcal{A}$.
- **Essentially surjective** if for all $B \in \mathcal{B}$, there exists an $A \in \mathcal{A}$ such that $F(A) \cong B$.

This gives the following equivalent definition of an equivalence of categories, which we will be using.

Proposition 3.1.6. *A functor $F : \mathcal{A} \to \mathcal{B}$ is an equivalence of categories if and only if F is full, faithful and essentially surjective.*

Proof. Checking that an equivalence satisfies the three properties is straightforward. For the other direction, given a fully faithful functor F that is essentially surjective, one chooses for each element $B \in \mathcal{B}$ an element $A \in \mathcal{A}$ and isomorphism $\eta_B : F(A) \to B$ (this uses essential surjectivity), and defines $G(B) = A$. One checks that G is functorial, and that η is a natural transformation $F \circ G \to 1_{\mathcal{B}}$. □

3.2 An Equivalence of Categories

We are now ready to state what is now the standard, modern formulation of the Deuring correspondence, due to Kohel [47, Chapter 5.3]. We follow [78, Chapter 42] closely, reformulating slightly to work only with (integral) ideals, as this closer matches how one computes with the Deuring correspondence in practice.

For the remainder of this section, fix an odd prime p. We will also fix an elliptic curve $E_0/\overline{\mathbb{F}}_p$, and a quaternion order $\mathcal{O}_0 \subseteq B_{p,\infty}$, such that $\mathrm{End}(E_0) \cong \mathcal{O}_0$. Throughout this section, we will implicitly fix an isomorphism $\mathrm{End}(E_0) \cong \mathcal{O}_0$, and associate these objects through this isomorphism.

The first category we consider will be the category whose objects are pairs (E, φ), where E is a supersingular elliptic curve over $\overline{\mathbb{F}}_p$, and $\varphi : E_0 \to E$ is an isogeny. The morphisms in this category $\psi : (E, \varphi) \to (E', \varphi')$ are simply isogenies $\psi : E \to E'$, i.e. $\mathrm{Hom}((E, \varphi), (E', \varphi')) = \mathrm{Hom}(E, E')$, where the latter is the homsets in the category of supersingular elliptic curves over $\overline{\mathbb{F}}_p$. Denote this category by $\mathcal{SS}_{E_0,p}$

Remark 7. *The reason for considering the objects as pairs (E, φ), instead of simply E, is just to get a natural choice of integral ideals of $\mathcal{O}_0$ later. This is purely syntactical; one can easily check that the forgetful functor from $\mathcal{SS}_{E_0,p}$ to $\mathcal{SS}_p$ sending (E, φ) to E is an equivalence of categories.*

For the second category, fix a maximal order $\mathcal{O}_0 \subseteq B_{p,\infty}$. We then consider a natural subcategory of the category of left $\mathcal{O}_0$-modules, namely the category of (integral) left $\mathcal{O}_0$-ideals, still under homomorphisms of $\mathcal{O}_0$-modules, denoted as $\mathsf{ideals}\ \mathcal{O}_0$.

Through the fixed isomorphism $\mathrm{End}(E_0) \cong \mathcal{O}_0$, recall that there is a natural association of left $\mathcal{O}_0$-ideals to isogenies given by Definition 2.2.8. We now wish to construct a dual of this operation, i.e. we wish to associate isogenies from E_0 to $\mathcal{O}_0$-ideals. This is obtained by the following lemma.

Lemma 3.2.1. *Let $\varphi : E_0 \to E$ be an isogeny. Then*

$$I_\varphi := \{\psi \circ \varphi \mid \psi \in \mathrm{Hom}(E, E_0)\}$$

is an (integral) left $\mathcal{O}_0$-ideal. Further, we have that

$$\mathrm{End}(E) \cong \mathcal{O}_R(I).$$

Proof. It is clear that $I_\varphi \subseteq \mathcal{O}_0$ is closed under addition. Further since $\psi I_\varphi \subseteq I_\varphi$ for all $\psi \in \mathcal{O}_0$, it is a left $\mathcal{O}_0$-ideal. Finally, by Proposition 2.2.3, φ gives an embedding

$$\varphi^* : \mathrm{End}(E) \hookrightarrow \mathrm{End}_{\mathbb{Q}}(E_0),$$
$$\varphi^*(\theta) = \frac{1}{\deg \varphi} \widehat{\varphi} \theta \varphi.$$

Thus, we see that $\mathrm{End}(E) \hookrightarrow \mathcal{O}_R(I_\varphi)$, since given $\theta \in \mathrm{End}(E)$ and $\psi\varphi \in I_\varphi$, we have that

$$\psi\varphi\varphi^*(\theta) = \psi\varphi \frac{1}{\deg \varphi} \widehat{\varphi}\theta\varphi = \psi\theta\varphi \in I_\varphi.$$

But by Theorem 2.3.2, $\mathrm{End}(E)$ is maximal, thus $\mathrm{End}(E) = \mathcal{O}_R(I_\varphi)$. □

The ideal I_φ in Lemma 3.2.1 will often be referred to as the **kernel ideal** of φ. The following proposition shows that the kernel ideal is a dual operation to taking the isogeny defined by an ideal.

Proposition 3.2.2. *Let I be a left $\mathcal{O}_0$-ideal, and let φ be an isogeny.*

(i) $I_{\varphi_I} = I$,

(ii) $\rho \circ \varphi_{I_\varphi} = \varphi$, *for some isomorphism ρ.*

Proof. See [78, Proposition 42.2.16 (b)], and [78, Corollary 42.2.21]. □

Next, we consider isogenies $\eta : E \to E'$, and show that these induce homomorphisms between the associated ideals.

Lemma 3.2.3. *Let $(E, \varphi), (E', \varphi') \in \mathcal{SS}_p$, and let $\eta : E \to E'$ be any isogeny. The map*

$$\eta^{\#} : I_{\varphi'} \to I_{\varphi}$$

defined by

$$\eta^{\#}(\psi \circ \varphi') = \psi \circ \eta \circ \varphi$$

is a homomorphism of left $\mathcal{O}_0$-modules.

Proof. Clearly, for $\psi \in \mathrm{Hom}(E, E_0)$, we have that $\psi \circ \hat{\eta} \in \mathrm{Hom}(E', E_0)$, hence $\eta^{\#}$ maps I_{φ} to $I_{\varphi'}$. Further, for $\psi_1, \psi_2 \in \mathrm{Hom}(E, E_0)$ and $\alpha \in \mathcal{O}_0$, checking that

$$\eta^{\#}(\alpha(\psi_1\varphi + \psi_2\varphi)) = \alpha(\eta^{\#}(\psi_1\varphi) + \eta^{\#}(\psi_2\varphi))$$

is straight-forward, hence $\eta^{\#} : I_{\varphi} \to I_{\varphi'}$ is a left $\mathcal{O}_0$-module homomorphism. □

Lemma 3.2.3 showed that $\eta \in \mathrm{Hom}((E, \varphi), (E', \varphi'))$ induced maps $\eta^{\#} \in \mathrm{Hom}(I_{\varphi}, I_{\varphi'})$. Finally, the last piece shows that each isogeny gives a unique $\mathcal{O}_0$-module homomorphism, and all $\mathcal{O}_0$-module homomorphisms arise this way, for which we again refer to Voight [78].

Lemma 3.2.4. *Let $(E, \varphi), (E', \varphi') \in \mathcal{SS}_p$. Then, the function*

$$\begin{gathered}\mathrm{Hom}(E, E') \to \mathrm{Hom}(I_{\varphi}, I_{\psi}),\\ \eta \to \eta^{\#}\end{gathered}$$

is a bijection.

Proof. See [78, Lemma 42.2.22], which shows that there is a natural bijection between $\mathrm{Hom}(E, E')$ and $\mathrm{Hom}(\mathrm{Hom}(E, E_0), \mathrm{Hom}(E', E_0))$, and combine this with the fact that $\mathrm{Hom}(E, E_0) \cong I_{\varphi}$ as $\mathcal{O}_0$-modules [78, Lemma 42.2.7]. □

These are all the pieces we need.

Theorem 3.2.5. *The functor $F : \mathcal{SS}_p \to \textsf{ideals}\,\mathcal{O}_0$ defined by sending objects to*

$$F((E, \varphi)) = I_{\varphi}$$

and morphisms $\eta \in \mathrm{Hom}((E, \varphi), (E', \varphi'))$ to

$$F(\eta) = \eta^{\#} : I_{\varphi} \to I'_{\varphi}$$

as in Lemma 3.2.3, is an equivalence of categories.

Proof. First, note that F is a functor: By Lemma 3.2.1, I_φ is an $\mathcal{O}_0$-ideal, and by Lemma 3.2.3, $\eta^\#$ is a homomorphism of left $\mathcal{O}_0$-modules. Further, given $\eta \in \operatorname{Hom}((E,\varphi),(E',\varphi'))$, and $\eta' \in \operatorname{Hom}((E',\varphi'),(E'',\varphi''))$, it is straight-forward to verify that

$$\eta^\# \circ \eta'^\# = (\eta \circ \eta')^\#,$$

hence F is indeed a functor.

We have already show that F is an equivalence; given any left $\mathcal{O}_0$-ideal I, Proposition 3.2.2 shows that $F((E_I,\varphi_I)) = I_{\varphi_I} = I$, hence F is essentially surjective and by Lemma 3.2.4 F is fully faithful, finishing the proof. □

The Deuring correspondence allows us to translate between the concepts from Chapter 1 and Chapter 2. We summarize in the corollary below, whose statements were either proved throughout this chapter, or follow immediately from the Deuring correspondence.

Corollary 3.2.6. *The following hold*

1. $\deg \varphi_I = \operatorname{nrd}(I)$, *and* $\operatorname{nrd}(I_\varphi) = \deg \varphi$.
2. $\widehat{\varphi_I} = \varphi_{\bar{I}}$ *and* $\bar{I}_\varphi = I_{\hat{\varphi}}$.
3. $\varphi_{IJ} = \varphi_J \circ \varphi_I$.
4. $\operatorname{End}(E_I) = \mathcal{O}_R(I)$ *and* $\operatorname{Aut}(E_I) = \mathcal{O}_R(I)^\times$.
5. $I \sim_R J \Rightarrow \operatorname{End}(E_I) \cong \operatorname{End}(E_J)$.
6. $\operatorname{End}(E_I) \cong \operatorname{End}(E_J) \Rightarrow I \sim_R J$ *or* $I \sim_R JP$, *where P is the unique ideal of norm p in $\mathcal{O}_R(I)$.*

We can also state several other corollaries about supersingular elliptic curves. The first is the original formulation of the Deuring correspondence.

Corollary 3.2.7. *Let p be a prime. There is a bijection between isomorphism classes of supersingular elliptic curves over $\overline{\mathbb{F}}_p$ up to Galois conjugacy, and maximal orders in $B_{p,\infty}$.*

Proof. The statement follows directly from the last two points in Corollary 3.2.6, combined with the fact that $j(E) \in \mathbb{F}_p$ if and only if $\mathbb{Z}[\sqrt{-p}]$ embeds in $\operatorname{End}(E)$ if and only if the unique non-trivial two-sided ideal of $\mathcal{O}$ is principal (see Proposition 1.3.22). □

The Deuring correspondence also makes it easy to count the number of isomorphism classes of elliptic curves over $\overline{\mathbb{F}}_p$.

Corollary 3.2.8. *The number of isomorphism classes of supersingular elliptic curves over $\overline{\mathbb{F}}_p$ is $\lfloor \frac{p}{12} \rfloor + e$, where*

$$e = \begin{cases} 0 & \textit{if } p \equiv 1 \pmod{12} \\ 1 & \textit{if } p \equiv \pm 5 \pmod{12} \\ 2 & \textit{if } p \equiv 11 \pmod{12} \end{cases}$$

Proof. By combining the Eichler mass formula (Theorem 1.3.17) with Theorem 3.2.5, we know that

$$\sum_{[E] \text{supersingular}} \frac{1}{\# \operatorname{Aut}(E)} = \sum_{[J] \in \operatorname{Cls} \mathcal{O}} \frac{1}{\# \mathcal{O}_R(J)^\times} = \frac{p-1}{24},$$

from which the result follows by combining with the size of $\operatorname{Aut}(E)$ given in Corollary 2.2.6, and the fact that $j(E) = 0$ is supersingular if and only if $p \equiv 2 \pmod{3}$, and likewise, $j(E) = 1728$ is supersingular if and only if $p \equiv 3 \pmod{4}$. □

Finally, the following can be seen as a special case of Tate's isogeny theorem, which states that two curves are isogenous if and only if they have the same number of rational points (which follows from [75]). However, accepting the Deuring correspondence, this special case becomes an almost trivial corollary.[1]

Corollary 3.2.9. *Let $E/\overline{\mathbb{F}}_p$ and $E'/\overline{\mathbb{F}}_p$ be two supersingular elliptic curves. Then there exists an isogeny $\varphi : E \to E'$.*

Proof. By Proposition 1.3.19, there exists a connecting ideal between their endomorphism rings. □

We now turn to describe explicitly how to compute the Deuring correspondence. We will see that the main ideas behind the proof of the Deuring correspondence give rise to the algorithms for computing it too.

3.3 Computing the Deuring Correspondence

As a correspondence, there are naturally two directions we could consider computing. Working with the fixed curve E_0, and isomorphism $\operatorname{End}(E_0) \cong \mathcal{O}_0$, we first consider translating left $\mathcal{O}_0$-ideals to their corresponding pairs (E_I, φ_I), before describing how to translate a pair $(E, \varphi) \in \mathcal{SS}_{E_0,p}$ to its corresponding ideal.

[1] Admittedly, we still rely on [75], as it is hidden in some of the referenced proofs from this section.

3.3.1 The Efficient Direction and the KLPT Algorithm

Translating an ideal of suitable norm to an isogeny is done through a direct application of Definition 2.2.8, as the following example shows.

Example 15: Ideal to isogeny translation. Fix $p = 109$, and recall that we saw that[a]

$$E : y^2 = x^3 + 34x + 36$$

has endomorphism ring

$$\begin{aligned} \mathrm{End}(E) &= \mathbb{Z} + \iota\mathbb{Z} + (\iota \circ \pi)\mathbb{Z} + \frac{([2] \circ \iota - [1]) \circ ([1] - \pi)}{11}\mathbb{Z} \\ &\cong \mathbb{Z} + \frac{1 + \mathbf{i}}{2}\mathbb{Z} + \frac{\mathbf{j} + \mathbf{k}}{2}\mathbb{Z} + \frac{\mathbf{i} - \mathbf{k}}{11}\mathbb{Z}, \end{aligned}$$

where π was the p-power Frobenius and ι satisfied $([2] \circ \iota - [1])^2 = [-11]$.

Consider the ideal

$$\begin{aligned} I &= 5\mathbb{Z} + \frac{5 + 5\mathbf{i}}{2}\mathbb{Z} + (4 + \mathbf{j})\mathbb{Z} + \frac{44 + 34\mathbf{i} + 11\mathbf{j} + \mathbf{k}}{22}\mathbb{Z} \\ &\cong \mathrm{End}(E)\langle 5, 2 + \iota + 2\pi + \iota\pi \rangle. \end{aligned}$$

Of course, when using Definition 2.2.8, it is sufficient to find the kernels of a generating set of I, so setting $\alpha = 2 + \iota + 2\pi + \iota\pi$, we find

$$E[I] = \ker[5] \cap \ker \alpha = E[5] \cap \ker \alpha = \widehat{\alpha}(E[31]),$$

where the last equality is true, because α is primitive, and thus $\ker \alpha = \widehat{\alpha}(E[\mathrm{nrd}(\alpha)])$.

Fixing $\mathbb{F}_{p^2} = \mathbb{F}_p(\omega)$, where $\omega^2 = -11$, we find a basis

$$E[5] = \langle P, Q \rangle = \langle (78\omega + 39, 98\omega + 94), (89, 49\omega) \rangle.$$

We further compute

$$\begin{aligned} \iota(P) &= (91, 73\omega), \\ \pi(P) &= (31\omega + 39, 11\omega + 94), \\ \iota(\pi(P)) &= (82, 6), \\ \widehat{\alpha}(P) &= [3]P - \iota(P) - [2]\pi(P) - \iota(\pi(P)) = (91, 36\omega). \end{aligned}$$

Since $\widehat{\alpha}(P)$ generates a group of order 5, we conclude that

$$E[I] = \langle (91, 36\omega) \rangle.$$

Computing the corresponding isogeny using e.g. Vélu gives

$$\varphi_I : E \to E_I,$$

where

$$E_I : y^2 = x^3 + 88x + 66,$$

and by the Deuring correspondence, we know that

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

[a] Again, notice that $x^3 + 34x + 36 = x^3 - 1056x + 13552$ in $\mathbb{F}_{109}[x, y]$.

This strategy is clearly not efficient for every ideal. Assuming $\mathrm{nrd}(I) = N$, where N is say a large prime, the points in $E[N]$ might be defined over exponentially large extension fields, and furthermore, the computation of N-isogenies from Velu-like algorithms is not efficient.

The key to solving this is by applying the KLPT algorithm [48], which roughly speaking takes an ideal I, and solves the related norm form (see Section 1.3.3)

$$q_I(\alpha) = T,$$

where T is sufficiently large and not prime. When applying the KLPT algorithm to the problem of translating a quaternion ideal I to its corresponding isogeny

$$\varphi_I : E_0 \to E_I,$$

one chooses some suitable number T, so that the computation of T-isogenies is efficient (i.e. so that T is smooth, and all for all prime powers $\ell_i^{r_i} \mid T$, $E[\ell_i^{r_i}]$ is defined over a reasonably small extension field). Then, given $\alpha \in I$ solving the norm equation, one uses the map from Lemma 1.3.23 and sets

$$J := \chi_I(\alpha),$$

and translates J to

$$\varphi_J : E_0 \to E_J$$

instead. Since I is equivalent to J, we have $E_J \cong E_I$ by the Deuring correspondence.[2]

There are several caveats to this story that make the situation more complicated. First, the large norm requirement on T makes it harder to choose a T such that this is always efficient. The basic choice is choosing T to be power smooth, i.e. satisfying $\ell_i^{r_i} \mid T \Rightarrow \ell_i^{r_i} \leq B$ for as small a bound B as obtainable. However, in Paper 1 we show

[2] For many applications, this is sufficient. However, if one really wants φ_I and not just E_I, this can also be computed from φ_J and α.

that this is far from optimal in practice, and discuss better choices, which makes this strategy more practical for larger parameters.

Second, the basic KLPT algorithm only works for $\mathcal{O}_0$-ideals, where $\mathcal{O}_0$ is of a very special form. If I is a connecting $(\mathcal{O}_1, \mathcal{O}_2)$-ideal, then a basic strategy is to find two ideals, a connecting $(\mathcal{O}_1, \mathcal{O}_0)$-ideal J_1 and a connecting $(\mathcal{O}_0, \mathcal{O}_2)$-ideal J_2 such that I is equivalent to $J_1 J_2$. This can again be done straightforwardly with the KLPT algorithm, however, in some cases (mainly cryptographic applications), this is not sufficient, and one has to apply more advanced generalisations of the KLPT algorithm. For an excellent overview of different KLPT-like algorithms, as well as detailed ideal-to-isogeny algorithms with different applications, we refer to Leroux's thesis [54].

3.3.2 The Hard Direction

When given an isogeny $\varphi : E \to E_1$, where E has known endomorphism ring, translating φ to its corresponding ideal I_φ can be done efficiently by again using Definition 2.2.8, but this time in reverse. More accurately, if φ is a cyclic isogeny of degree N, and

$$\mathrm{End}(E) = \alpha_1\mathbb{Z} + \alpha_2\mathbb{Z} + \alpha_3\mathbb{Z} + \alpha_4\mathbb{Z},$$

then one fixes a basis $E[N] = \langle P, Q \rangle$, and evaluates α_i on this basis to find its action matrix $\mu_{\alpha_i} \in \mathbf{M}_2(\mathbb{Z}/\mathbb{Z})$, i.e.

$$\mu_{\alpha_i} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

means that $\alpha(P) = [a]P + b[Q]$ and $\alpha_i(Q) = [c]P + [d]Q$. Then, if $\ker\varphi$ is generated by $[y_1]P + [y_2]Q$, one looks for a solution x_1, x_2, x_3, x_4 of the linear system

$$\sum_{1 \le i \le 4} x_i \mu_{\alpha_i} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = 0,$$

such that $\sum_{1 \le i \le 4} x_i \mu_{\alpha_i} \in \mathbf{GL}_2(\mathbb{Z}/N\mathbb{Z})$. Given such a solution, it is clear that the ideal

$$I_\varphi = \mathrm{End}(E)\langle N, x_1\alpha_1 + x_2\alpha_2 + x_3\alpha_3 + x4\alpha_4 \rangle$$

is the ideal we are after, by Proposition 3.2.2.

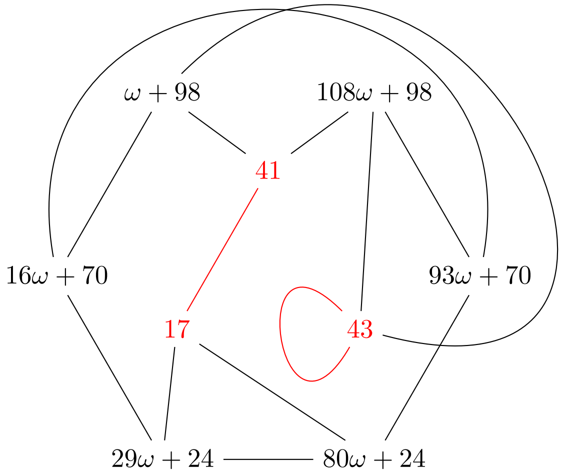
However, this already required that $\mathrm{End}(E)$ was known.[3] Computing $\mathrm{End}(E)$ is known as **the endomorphism ring problem**, a problem which more or less all of isogeny-based cryptography seems to rely on being hard, and which was proven to be equivalent

[3] Note that this was not required in the other direction: Given an ideal I, the first step of translating I to its corresponding isogeny would typically be to find E with $\mathrm{End}(E) \cong \mathcal{O}_R(I)$. This can always be found, by translating any correcting $(\mathcal{O}_0, \mathcal{O}_R(I))$-ideal, and where a choice of curve E_0 and $\mathcal{O}_0$ with $\mathrm{End}(E_0) \cong \mathcal{O}_0$ can always be found, by computing a supersingular reduction of a CM curve over $\mathbb{Q}$ [7]. See Paper 1 which goes over this in much greater detail.

(assuming the generalised Riemann-hypothesis) to the isogeny path-finding problem by Wesolowski [80]. See also the excellent overview of reductions between common hardness assumptions in isogeny-based cryptography by Wesolowski [81].

The best-known algorithm for computing the endomorphism ring is using the Delfs–Galbraith algorithm for computing an isogeny-path to a curve with known endomorphism ring [35], which runs in $\widetilde{O}(p^{\frac{1}{2}})$. With a quantum computer, Biasse, Jao, and Sankar [6] showed that this can be improved to $\widetilde{O}(p^{\frac{1}{4}})$. The Delfs–Galbraith algorithm is essentially an algorithm for finding isogeny paths between supersingular curves defined over $\mathbb{F}_p$, using their $\mathbb{Z}[\sqrt{-p}]$-orientation, which runs in $\widetilde{O}(p^{\frac{1}{4}})$. Thus, for finding isogeny-paths between supersingular curves defined over $\mathbb{F}_{p^2}$, one can do a random walk from both curves, until one hits curves that happen to be defined over $\mathbb{F}_p$, before applying Delfs–Galbraith.

Example 16: 2-isogeny graph. Let $p = 109$, and let $\mathbb{F}_{p^2} = \mathbb{F}_p(\omega), \omega^2 = -11$. We draw the 2-isogeny graph over $\overline{\mathbb{F}}_p$, marking vertices by their j-invariant.



The curves and isogenies defined over $\mathbb{F}_p$ are marked in red. At first, the graph seems to contradict Theorem 2.3.7, since $\#\,\mathrm{Cl}(\mathbb{Z}[\sqrt{-109}]) = 6$, and there are only three j invariants over $\mathbb{F}_p$, however, this has a natural explanation: Theorem 2.3.7 considers oriented curves up to *oriented* isomorphism. For every curve over $\mathbb{F}_p$, we get two $\mathbb{Z}[\sqrt{-109}]$-orientations by sending $\sqrt{-109}$ to π and $-\pi$ respectively (if one wishes to keep the orientation fixed, one can instead consider a curve and its quadratic twist, which are isomorphic over $\overline{\mathbb{F}}_p$, but not $\mathbb{F}_p$, which exactly corresponds to an isomorphism which is not $\mathbb{Z}[\sqrt{-p}]$-oriented).

Notice also that 2 is ramified in $\mathbb{Z}[\sqrt{-109}]$. In particular, this means that given an ideal $\mathfrak{l}$ above 2, $\mathfrak{l} = \overline{\mathfrak{l}}$, and thus there is a unique horizontal $\mathbb{F}_p$-rational isogeny of degree 2 from each curve over $\mathbb{F}_p$. This also explains the weird phenomenon happening at $j(E) = 43$, which at first glance seems to have more 2-isogenies than the other curves. This is false: We have drawn the graph undirected, associating isogenies and their duals; however, the oriented 2-isogeny from $j(E) = 43$ is self-dual.

We also point out that the vertices in the volcanoes in Example 11 surjects onto the vertices in the graph in this example, by taking the j-invariants of the oriented curves (and the edges surjects onto the corresponding edges in the 3-isogeny graph). In this sense, one can use orientations to "give more structure to" the usual isogeny-graph.

By Corollary 3.2.8, the total number of supersingular curves is $O(p)$, while by Theorem 2.3.7[4], the number of supersingular curves over $\mathbb{F}_p$ is $O(p^{\frac{1}{2}})$. Using more properties of supersingular isogeny-graphs, one can show that the random walk to an $\mathbb{F}_p$-curve is expected to take $O(p^{\frac{1}{2}})$ steps.

There are also direct algorithms for computing endomorphism rings by Eisenträger, Hallgren, Leonardi, Morrison, and Park [38] (later improved in Fuselier, Iezzi, Kozek, Morrison, and Namoijam [41]).

3.4 Optimal Embeddings and Orientations

Considering optimal embeddings through the Deuring correspondence leads to the beautiful theory of oriented supersingular curves (see Section 2.3.2). This theory is usually derived through supersingular reduction of CM curves. However, in this section, we restate a few results from Paper 4, and expand on these, to deduce the analogue of Theorem 2.3.7, on the quaternion side. To the best of our knowledge, this is a novel proof of the main theorem in this section.

For the rest of this section, assume that $(\mathcal{O}, \iota)$ is a primitivel $\mathfrak{O}$-oriented order, i.e. that $\iota\,|_{\mathfrak{O}} \colon \mathfrak{O} \hookrightarrow \mathcal{O}$ is an optimal embedding.

3.4.1 Quaternion Ideals Generated by Quadratic Ideals

We first restate some easy results from Paper 4.

Proposition 3.4.1. *Let $(\mathcal{O}, \iota)$ be a primitively $\mathfrak{O}$-oriented order. Then*

[4] Recall that the curves defined $\mathbb{F}_p$ are precicely the ones admitting an $\mathbb{Z}[\sqrt{-p}]$-orientation.

- *Given a left $\mathcal{O}$-ideal I, we have that $\mathcal{O}\langle \mathrm{nrd}(I)\rangle \subseteq \mathcal{O}\langle I \cap \iota(K)\rangle \subseteq I$.*
- *Given an invertible $\mathfrak{O}$-ideal $\mathfrak{l}$, we have that $\mathcal{O}\langle \iota(\mathfrak{l})\rangle \cap \iota(K) = \iota(\mathfrak{l})$.*

Proof. See Paper 4, Proposition 6. □

The proposition above motivates the following definition.

Definition 3.4.2. Let $(\mathcal{O}, \iota)$ be a primitively $\mathfrak{O}$-oriented maximal order. A left $\mathcal{O}$-ideal is said to be **generated by an $\mathfrak{O}$-ideal** if

$$I = \mathcal{O}\langle I \cap \iota(K)\rangle.$$

When ideal a left $\mathcal{O}$-ideal I is generated by an $\mathfrak{O}$-ideal, then $\mathcal{O}_R(I)$ is automatically also primitively $\mathfrak{O}$-oriented. Moreover, this orientation agrees with the one on $\mathcal{O}$, as the following lemma shows.

Lemma 3.4.3. *Let $(\mathcal{O}, \iota)$ be a primitively $\mathfrak{O}$-oriented maximal order, let $\mathcal{O}'$ be another maximal order, and let I a connecting $(\mathcal{O}, \mathcal{O}')$-ideal. Then I is generated by an $\mathfrak{O}$-ideal if and only if $(\mathcal{O}', \iota)$ is a (not necessarily primitively) $\mathfrak{O}$-oriented order.*

Proof. See Paper 4, Lemma 8 and Lemma 9. □

3.4.2 Class Group Action on the Oriented Class Set

In Section 1.4, we defined the set of $\mathfrak{O}$-oriented ideal classes $\mathrm{Cls}_{\mathfrak{O}}(\mathcal{O})$. We now prove that there is a free action of $\mathrm{Cl}(\mathfrak{O})$ on this set, in one or two orbits.

Lemma 3.4.4. *Let $\mathfrak{O}$ be a quadratic order of conductor f. Given an $\mathfrak{O}$-ideal $\mathfrak{a}$, and $(I, \omega) \in \mathrm{Cls}_{\mathfrak{O}}(\mathcal{O})$, define*

$$\mathfrak{a} \star (I, \omega) := (IJ, \omega),$$

where $J = \mathcal{O}_R(I)\langle \iota_\omega(\mathfrak{a})\rangle$. This induces a well defined group action by $I(\mathfrak{O}, f)$ on $\mathrm{Cls}_{\mathfrak{O}}(\mathcal{O})$.

Proof. By Lemma 3.4.3, $(\mathcal{O}_R(\mathcal{O}_R(I)\langle \iota_\omega(\mathfrak{a})\rangle), \iota_\omega)$ is an oriented order, and primitively so, because $\mathfrak{a}$ is coprime to the conductor f, hence the first thing we need to show is that the element is well defined regardless of representative chosen.

When $(I, \omega) \sim (I', \omega')$, there is an element $\alpha \in B$ defining an ideal equivalence $I\alpha = I'$, in addition to satisfying $\alpha^{-1}\omega\alpha = \omega'$. Thus, writing $J = \mathcal{O}_R(I)\langle \iota_\omega(\mathfrak{a})\rangle$ and $J' = \mathcal{O}_R(I')\langle \iota_{\omega'}(\mathfrak{a})\rangle$, we have that

$$\begin{aligned} \alpha^{-1}J\alpha &= \alpha^{-1}\mathcal{O}_R(I)\langle \iota_\omega(\mathfrak{a})\rangle\alpha \\ &= \alpha^{-1}\mathcal{O}_R(I)\alpha\langle \alpha^{-1}\iota_\omega(\mathfrak{a})\alpha\rangle \\ &= \mathcal{O}_R(J)\langle \iota_{\omega'}(\mathfrak{a})\rangle = J'. \end{aligned}$$

From this it follows that $\alpha^{-1}IJ\alpha = \alpha^{-1}I\alpha\alpha^{-1}J\alpha = I'J'$, hence the element $\mathfrak{a} \star (I, \omega)$ is well defined regardless of equivalence class chosen.

Next, we show that this is in fact a group action by group $I(\mathfrak{O})$ on $\mathrm{Cls}_{\mathfrak{O}}$. Let $\mathfrak{a}, \mathfrak{b} \in I(\mathfrak{O})$, and write

$$I_1 := \mathcal{O}\langle \iota(\mathfrak{b})\rangle, \qquad I_2 := \mathcal{O}_R(I_1)\langle \iota(\mathfrak{a})\rangle, \qquad I := \mathcal{O}\langle \iota(\mathfrak{ab})\rangle.$$

It is easy to see that $I \subseteq I_1 \cdot I_2$, and by the multiplicativity of the norms (I_1 and I_2 are compatible by definition), we see that $\mathrm{nrd}(I) = \mathrm{nrd}(I_1 \cdot I_2)$, hence $I = I_1 \cdot I_2$, which again implies that

$$\mathfrak{a} \star (\mathfrak{b} \star (I, \omega)) = (\mathfrak{a} \star \mathfrak{b}) \star (I, \omega),$$

finishing the proof. □

From this lemma, we get the following theorem.

Theorem 3.4.5. *Let $\mathfrak{O} \subset K$ be an imaginary quadratic order. Then, $\mathrm{Cl}(\mathfrak{O})$ acts freely on $\mathrm{Cls}_{\mathfrak{O}}(\mathcal{O})$ in exactly $1 - \left(\frac{d_K}{p}\right)$ orbits, where the group action is defined by*

$$\begin{aligned} \mathrm{Cl}(\mathfrak{O}) \times \mathrm{Cls}_{\mathfrak{O}}(\mathcal{O}) &\to \mathrm{Cls}_{\mathfrak{O}}(\mathcal{O}) \\ [\mathfrak{a}] \star ([I], \omega) &\to (IJ, \omega), \end{aligned}$$

where $J = \mathcal{O}_R(I)\langle \iota_\omega(\mathfrak{a})$.

Proof. Given an element $(I, \omega) \in \mathrm{Cls}_{\mathfrak{O}}(\mathcal{O})$, we start by showing that the stabiliser of (I, ω) in $I(\mathfrak{O})$ are precicely the principal ideals $P(\mathfrak{O})$. Assume that $\mathfrak{a} = (\alpha)$ for some $\alpha \in K$. Then

$$\mathfrak{a} \star (I, \omega) = (I\iota_\omega(\alpha), \iota_\omega(\alpha)^{-1}\omega\iota_\omega(\alpha)) = (I\iota_\omega(\alpha), \omega),$$

where the last equality follows from the fact that $\iota_\omega(\alpha), \omega \in \iota_\omega(K)$ commute. Conversely, assume that $\alpha \star (I, \omega) = (J, \omega) \sim (I, \omega)$. By definition, this implies that there exists some $\alpha \in B$ such that $I\alpha = J$, and thus $\mathcal{O}_R(I)\langle \iota_\omega(\mathfrak{a})\rangle = \mathcal{O}_R(I)\alpha$. However, conjugation by α also leaves ι_ω fixed, hence we know that $\alpha = \iota_\omega(\beta)$ for some $\beta \in K$, thus $\mathfrak{a} = (\beta)$ is principal.

This shows that $\mathrm{Cl}(\mathfrak{O})$ acts freely on $\mathrm{Cls}_{\mathfrak{O}}$. The number of orbits comes from combining the orbit-stabilizer theorem with Theorem 1.4.6. □

The previous theorem could also have been given directly as a corollary of the Deuring correspondence and Theorem 2.3.7. However, we stress that everything we have done to prove this has been purely on the quaternion side: Thus, we can turn it around and point out that viewing Theorem 3.4.5 through the Deuring correspondence gives a new proof of Theorem 2.3.7.

Example 17: Class group action on ideal classes. Let $p = 109$, and let $B_{p,\infty} = (-11, -109 \mid \mathbb{Q})$. Let $K = \mathbb{Q}(\sqrt{-23})$, and let $\mathfrak{O}_K = \mathbb{Z}[\delta]$, where $\delta = \frac{1+\sqrt{-23}}{2}$ be the maximal order. Recall that we in Example 6, we computed the set $\mathrm{Cls}_{\mathfrak{O}_K}(\mathcal{O})$. We reuse the representatives chosen from that example.

As in previous examples, let

$$\mathfrak{l} = 3\mathbb{Z} + \frac{1-\sqrt{-23}}{2}\mathbb{Z},$$

and recall that that $\langle [\mathfrak{l}_3] \rangle = \mathrm{Cl}(\mathfrak{O}_K) \cong \mathbb{Z}/3\mathbb{Z}$. The action of $[\mathfrak{l}]$ on $\mathrm{Cls}_{\mathfrak{O}_K}(\mathcal{O})$ thus gives the two orbits by Theorem 3.4.5, as shown in Figure 3.1. Notice especially the importance of working with $\mathrm{Cls}_{\mathfrak{O}_K}(\mathcal{O})$ and not $\mathrm{Type}_{\mathfrak{O}_K}(\mathcal{O})$; Again, by Example 6, we know that

$$\left(O_R(I_7), \frac{11+12\mathbf{i}+\mathbf{k}}{22}\right) \cong \left(O_R(I_8), \frac{11-12\mathbf{i}-\mathbf{k}}{22}\right),$$

however, when acting by $[\mathfrak{l}_3]$, the codomains are no longer isomorphic.

Next, let $L = \mathbb{Q}(\sqrt{-109})$, and let $\mathfrak{O}_L = \mathbb{Z}[\sqrt{-109}]$. Recall that we in Example 6, we also computed the set $\mathrm{Cls}_{\mathfrak{O}_L}(\mathcal{O})$. We reuse the representatives chosen from that example. Now let

$$\mathfrak{l}_{11} = 11\mathbb{Z} + (1+\sqrt{-109})\mathbb{Z}.$$

Again, from Example 4, we have that $\langle \mathfrak{l}_{11} \rangle = \mathrm{Cl}(\mathbb{Z}[\sqrt{-109}])$, however, this time the action of is transitive: The single orbit is shown in Figure 3.2.

We also point out that the cycles in Figure 3.2 correspond exactly to the craters of the volcanoes in Example 13. For instance, writing

$$J = 5\mathbb{Z} + \frac{5+5\mathbf{i}}{2}\mathbb{Z} + (4+\mathbf{j})\mathbb{Z} + \frac{44+34\mathbf{i}+11\mathbf{j}+\mathbf{k}}{22}\mathbb{Z},$$

we can confirm that I_3 is equivalent to J as left ideals. In Example 15, we computed the isogeny corresponding to J under the Deuring correspondence, and found that

$$E_J : y^2 = x^3 + x^3 + 88x + 66,$$

which is again isomorphic to E_1 from Example 13 via the isomorphism

$$\begin{aligned} &\psi : E_J \to E_1 \\ &\psi(x, y) = ((-43\omega - 54)x, (26\omega - 26)y). \end{aligned}$$

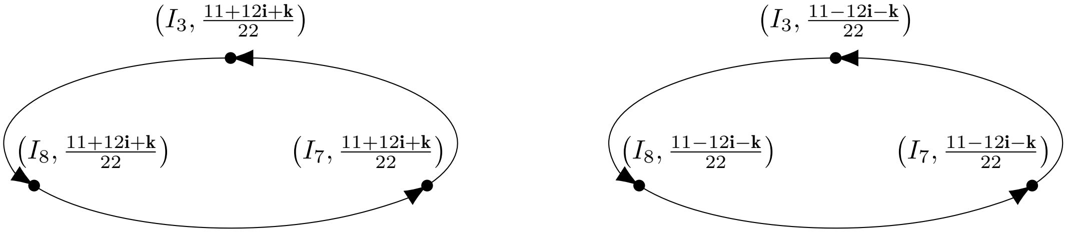

$\left(I_3, \frac{11+12\mathbf{i}+\mathbf{k}}{22}\right)$
$\left(I_8, \frac{11+12\mathbf{i}+\mathbf{k}}{22}\right)$
$\left(I_7, \frac{11+12\mathbf{i}+\mathbf{k}}{22}\right)$
$\left(I_3, \frac{11-12\mathbf{i}-\mathbf{k}}{22}\right)$
$\left(I_8, \frac{11-12\mathbf{i}-\mathbf{k}}{22}\right)$
$\left(I_7, \frac{11-12\mathbf{i}-\mathbf{k}}{22}\right)$

Figure 3.1: $[\mathfrak{l}_3] \in \mathrm{Cl}(\mathbb{Z}[\frac{1+\sqrt{-23}}{2}])$ acting on $\mathrm{Cls}_{\mathbb{Z}[\frac{1+\sqrt{-23}}{2}]}(\mathcal{O}_0)$. Under the Deuring correspondence, these cycles correspond exactly to the craters of the volcanoes in Example 13

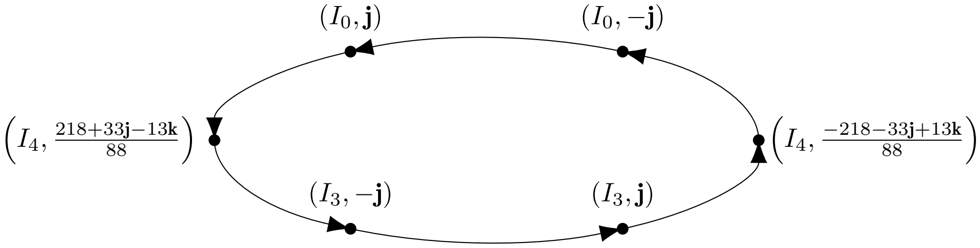

$(I_0, \mathbf{j})$
$(I_0, -\mathbf{j})$
$\left(I_4, \frac{218+33\mathbf{j}-13\mathbf{k}}{88}\right)$
$\left(I_4, \frac{-218-33\mathbf{j}+13\mathbf{k}}{88}\right)$
$(I_3, -\mathbf{j})$
$(I_3, \mathbf{j})$

Figure 3.2: $[\mathfrak{l}_{11}] \in \mathrm{Cl}(\mathbb{Z}[\sqrt{-109}])$ acting on $\mathrm{Cls}_{\mathbb{Z}[\sqrt{-109}]}(\mathcal{O}_0)$. The vertices correspond 2-to-1 with the marked vertices in Example 16.

Chapter 4

Cryptography from Endomorphism Rings

Finally, we end the background with a chapter on the cryptographic applications of the material in the previous chapters. We first give an abstract primitive, based on group actions. This construction generalises the classic Diffie–Hellman protocol, and we discuss how to instantiate this protocol with the theory of orientations. Next, we focus on a digital signature algorithm, SQIsign, which is based on the Deuring correspondence. Finally, we also give a brief overview of other topics in isogeny-based cryptography, which are less focal in this thesis.

4.1 Cryptographic Group Actions

In 1976, Diffie and Hellman introduced the first asymmetric cryptographic protocol [37], allowing two parties to arrive at a shared secret, communicating only over a public channel. Given an abelian group $G \cong \mathbb{Z}/N\mathbb{Z}$, the protocol is based on the following commutative diagram[1]

$$\begin{tikzcd}
G \arrow[r, "{[a]}"] \arrow[d, "{[b]}"'] & G \arrow[d, "{[b]}"] \\
G \arrow[r, "{[a]}"'] & G
\end{tikzcd}$$

[1] Although this is a non-traditional way of describing the Diffie–Hellman protocol, it will make the generalisation much clearer.

in the sense that the parties start with a public point $1_G \neq P \in G$, and choose their respective secret elements $a, b \in (\mathbb{Z}/N\mathbb{Z})^\times$.[2] By sending each other $[a]P$ and $[b]P$ respectively, they can both arrive at the shared secret $[a][b]P = [b][a]P$ in G.

The incredible simplicity and flexibility of the Diffie–Hellman protocol has made it a good starting point for countless other cryptographic protocols, such as signatures [42] more advanced key-management protocols [60]. However, the security relies[3] on the hardness of the **discrete logarithm problem**: given P and $[a]P$, find a. This problem has in fact known to be easy since 1994, when Peter Shor showed that there exists a polynomial time algorithm for solving the discrete logarithm problem [70]. The major caveat here is that Shor's algorithm only runs on a *quantum* computer.

Thus, we aim to generalise the Diffie–Hellman protocol, so that its security is no longer reliant on the discrete logarithm problem. The relatively simple idea is that all we really need for the Diffie–Hellman protocol to go through is a *group action* on a set, not necessarily all the structure of working in a group.

To this end, we follow the definition of Hard Homogenous Spaces by Couveignes [24] (which is very close to the definition of effective cryptographic group-action from [1]).

Definition 4.1.1. Let G be a finite, abelian group, and let G act freely and transitively on a (necessarily finite) non-empty set X. Then X is a **hard homogeneous space (HHS)** (for G) if the following are satisfied:

(i) The following problems are computationally easy:

Group Operation: Given $g_1, g_2 \in G$, compute $g_1^{-1}, g_1 g_2$, and decide if $g_1 = g_2$.

Random Sample: Sample a random $g \in G$ with uniform probability.

Membership: Decide if $x \in X$.

Equality: Given $x_1, x_2 \in X$, decide if $x_1 = x_2$.

Action: Given $g \in G, x \in X$, compute $g \star x$.

(ii) The following problems are computationally hard:

Vectorization: Given $x_1, x_2 \in X$, compute $g \in G$ such that $g \star x_1 = x_2$.

Parallellization: Given $x_1, g \star x_1, x_2 \in X$, compute $g \star x_2$.

Further, we say that X is a **very hard homogeneous space (VHHS)** if the following problem is also computationally hard:

[2] We are imposing an extra requirement here that a, b are invertible.

[3] Do not confuse the implications here: If the discrete logarithm problem is easy, then Diffie–Hellman is broken. However, it is not known whether breaking Diffie–Hellman implies that the discrete logarithm problem is easy. Depending on the security model, breaking Diffie–Hellman is *equivalent* to the **CDH problem** or the **DDH problem**.

ParallelTesting: Let $x_1, x_2, x_3, x_4 \in X$ be an output of one of the following two distributions, with equal probability:

- Sample a random $x_1, x_2 \in X$ and a random $g \in G$, and output $(x_1, g \star x_1, x_2, g \star x_2)$.
- Sample a random $x_1, x_2, x_3 \in X$ and a random $g \in G$, and output $(x_1, g \star x_1, x_2, x_3)$.

Decide which distribution (x_1, x_2, x_3, x_4) was sampled from, with a non-negligible advantage.

The following example shows that (very) hard homogeneous spaces really generalise the classic Diffie–Hellman situation.

Example 18: A (not so) HHS. Let $E/\mathbb{F}_p$ be an elliptic curve, such that $E(\mathbb{F}_p) \cong \mathbb{Z}/q\mathbb{Z}$ for some cryptographically large prime q. Then $(\mathbb{Z}/q\mathbb{Z})^\times$ acts freely and transitively on the *set* $X = \#E(\mathbb{F}_p) \setminus \{\infty\}$, where the group action is defined by

$$\begin{aligned}(\mathbb{Z}/q\mathbb{Z})^\times \star X &\to X \\ [n] \star P &= [n]P.\end{aligned}$$

It is immediately clear that all of the "easy" problems are in fact easy. Still, it is interesting to comment on the group action evaluation: In Example 4 we pointed at that for uniformly sampled $[n] \in (\mathbb{Z}/q\mathbb{Z})^\times$, and $P \in X$, computing $[n]P$ is easy precisely because of the double-and-add algorithm.

However, the hard problems are of course only conjecturally (classically) hard. Vectorization says that given $P, Q \in E(\mathbb{F}_p)$, we are tasked with computing $[d] \in \mathbb{Z}/q\mathbb{Z}$ such that $[d]P = Q$. This is precisely the discrete logarithm problem. Further, consider parallelization: Given $P, [d]P, Q \in E(\mathbb{F}_p)$, computing $[d]Q$ is precisely the CDH problem. Likewise, it is clear that ParallelTesting is precisely the DDH problem.

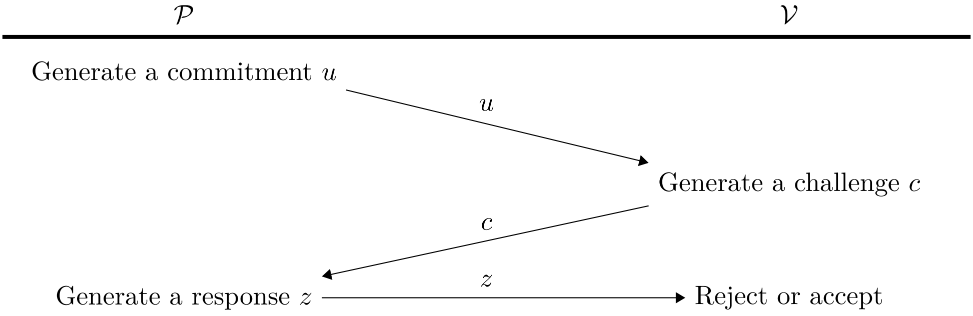
And finally, as pointed out, neither of these problems are hard for quantum computers.

Given a hard homogeneous space X for a group G, we can thus repeat the Diffie–Hellman construction, creating a key exchange based on the commutative diagram

$$\begin{tikzcd} X \arrow[r, "g_a \star -"] \arrow[d, "g_b \star -"'] & X \arrow[d, "g_b \star -"] \\ X \arrow[r, "g_a \star -"'] & X \end{tikzcd}$$

and proceeding exactly as before.

Σ-protocol. Next, we describe a simple Σ-protocol, which will serve as motivation for the last part of this section. Recall that a **Σ-protocol** is an interactive protocol, where a prover $\mathcal{P}$ wishes to convince a verifier $\mathcal{V}$ that he knows some secret value x corresponding to a public value y, without giving any information about x.



A classical example is such a Σ-protocol is proof of knowledge of discrete logarithm, due to Schnorr [67]. Below, we give a proof of knowledge of vectorization, again due to Couveignes [24]. For this, we need X to be a HHS for G, and $x_0 \in X$ to be a public starting element.

Setup: $\mathcal{P}$ randomly samples a secret $g_s \in G$, and sets her public value as $y := g_s \star x_0$.

Commitment: The Prover randomly samples $g_c \in G$, and commits to $u := g_c \star x_0$.

Challenge: The Verifier chooses a random challenge bit $b \in \{0, 1\}$.

Response: The Prover responds with $g := g_c g_s^{-b}$.

Verification: If $b = 0$, $\mathcal{V}$ checks that $g \star x_0 = u$; if $b = 1$, $\mathcal{V}$ checks that $g \star y = u$.

One easily verifies that if all parties act honestly, the verifier accepts. Further, assuming that the prover can answer both challenges correctly, i.e. the prover can compute $g_0 = g_c$ and $g_1 = g_c g_s^{-1}$, she must know the secret since she can recover it by $g_s = g_0/g_1$. Thus, assuming she does not know the secret, she must guess the challenge bit before committing, to get the verifier to accept. She has $\frac{1}{2}$ chance of guessing the bit correctly, and hence convincing the verifier that she knows the secret; by repeating the protocol n times, this can be lowered to $\frac{1}{2^n}$. Finally, the verifier learns nothing about x from the protocol. To see this, note that by choosing a challenge bit b first, the verifier can generate a transcript of the protocol that is indistinguishable from a real execution of the protocol.

The applications of Σ-protocols are vast. As a simple, but common example, one can obtain a digital signature algorithm from a Σ-protocols via the Fiat–Shamir heuristic [39].

4.1.1 CRS

What is typically referred to as the first example of a cryptosystem based on a hard problem related to the computation of isogenies, is the Couveignes-Rostovtsev-Stolbunov (CRS) scheme, which is a common term for two related schemes, independently created by Couveignes [25, 24] and Rostovtsev and Stolbunov [66, 73, 72]).

Formulated in terms of Hard Homogenous Spaces, the CRS scheme can be understood immediately from the material in Section 2.2.2: Letting $K = \mathbb{F}_p$, and letting $\mathfrak{O}$ be an imaginary quadratic order, $\mathcal{E}\ell\ell_K(\mathfrak{O})$ is conjecturally a HHS for $\mathrm{Cl}(\mathfrak{O})$. This largely follows from Theorem 2.2.10, although with a few caveats which we will continue to discuss now.

Group action: Let $E \in \mathcal{E}\ell\ell_K(\mathfrak{O})$, and assume for simplicity that $\mathbb{Z}[\pi] = \mathfrak{O}$, where π denotes the Frobenius endomorphism.[4] Given $[\mathfrak{l}] \in \mathrm{Cl}(\mathfrak{O})$, where $\mathfrak{l} = (\ell, \pi - \lambda)$, we start by computing

$$E[\mathfrak{l}] = E[\ell] \cap E[\pi - \lambda],$$

where this can again be found by evaluating π on $E[\ell]$. Finally, one computes $\varphi_{\mathfrak{l}}$ from $E[\mathfrak{l}]$ using Vélu.

It is immediately clear that $E[\ell]$ needs to be defined over a reasonable field extension, which already puts heavy restrictions on $\mathrm{nrd}(\mathfrak{l})$, and which is something that leads to CRS generally being too inefficient to be considered practical. Combining this with the fact that we need Vélu to be efficient, we arrive at the conclusion that $\mathrm{nrd}(\mathfrak{l})$ should at least be reasonably smooth. This can be done by choosing a smooth representative of $[\mathfrak{l}] \in \mathrm{Cl}(\mathfrak{O})$, which can be done by working modulo a lattice of relations, as in Section 1.2.4 (see specifically Example 4).

Hard problems: As in classical Diffie–Hellman, it is clear that for Hard Homogeneous Spaces, ParallelTesting reduces to Parallelization, which again reduces to Vectorization, and Childs, Jao, and Soukharev [16] showed that vectorization is solvable in quantum subexponential time with Kuperberg's algorithm [50]. While this still means that the problem is considered hard[5], it does force the parameters to be quite big, leading to even larger slowdowns, which makes the original CRS scheme seem impractical, without any fundamental improvements.

4.1.2 CSIDH

Although several efficiency improvements to the CRS scheme were made by De Feo, Kieffer, and Smith [29], the first fundamental change to the CRS scheme came with the introduction of CSIDH by Castryck, Lange, Martindale, Panny, and Renes [9], which led to a massive speed-up, and a somewhat practical version of the CRS scheme. Their idea

[4] In general, we only have $\mathbb{Z}[\pi] \subseteq \mathrm{End}(E) = \mathfrak{O}$.

[5] Compare with RSA in classical cryptography, and the sub-exponential factoring algorithms.

was to use supersingular curves over $\mathbb{F}_p$, together with the $\mathbb{Z}[\sqrt{-p}]$-orientation induced by π, where π again denotes the Frobenius endomorphism.[6] Thus, letting $\mathfrak{O} = \mathbb{Z}[\pi]$, CSIDH uses the Hard Homogeneous Space $SS_{\mathfrak{O}}^{\mathrm{pr}}(p)$ for $\mathrm{Cl}(\mathfrak{O})$, and otherwise works exactly like the CRS protocol.

Group action: The massive efficiency gain for CSIDH compared to CRS comes from the fact that the order of $E(\mathbb{F}_p)$ is always $(p+1)^2$ for supersingular curves (see Proposition 2.3.3). Thus, let

$$p = \prod_{\ell_i \in P} \ell_i - 1,$$

where P contains the first r primes. For $\ell \in P$, we have that ℓ is split in $\mathbb{Q}(\sqrt{-p})$, and further, it factors as $\ell = \mathfrak{l}\bar{\mathfrak{l}}$, for $\mathfrak{l} = (\ell, \pi - 1)$, and $\bar{\mathfrak{l}} = (\ell, \pi + 1)$, where $\pi^2 = -p$ (since $\ell \mid p + 1 = \mathrm{nrd}(\pi \pm 1)$). Thus

$$E[\mathfrak{l}] = E[\ell] \cap E[\pi - 1] = E[\ell] \cap E(\mathbb{F}_p),$$

i.e. $E[\mathfrak{l}]$ is the unique $\mathbb{F}_p$-rational subgroup of order ℓ. Similarly, one can show that $E[\bar{\mathfrak{l}}]$ is generated by the unique $\mathbb{F}_p$-rational point of order ℓ on E^t, a quadratic twist of E. Hence, it is clear that given an element $[\mathfrak{a}] \in \mathrm{Cl}(\mathfrak{O})$, where

$$\mathrm{nrd}(\mathfrak{a}) = \prod_{\ell_i \in P} \ell_i^{r_i},$$

we can evaluate $[\mathfrak{a}] \star E$ in reasonable time, as long as the r_i are small in absolute value. Heuristically, any element $[\mathfrak{b}] \in \mathrm{Cl}(\mathfrak{O})$ has a representative of this form; again, we find it by working modulo a lattice of relations.

The remaining problem: To truly satisfy the properties of a (V)HHS, we need to be able to evaluate $[\mathfrak{a}] \star E$, for $[\mathfrak{a}]$ uniformly sampled in $\mathrm{Cl}(\mathfrak{O})$. As we have explained, one does this by "smoothing" $\mathfrak{a}$ over the lattice of relations. However, computing the lattice of relations clearly implies computing the structure of $\mathrm{Cl}(\mathfrak{O})$; a problem for which there currently exists no polynomial-time classical algorithms.

For the smallest CSIDH-security parameter, the structure of $\mathrm{Cl}(\mathfrak{O})$ was computed by Castryck, Lange, Martindale, Panny, and Renes [9] in a world-record class group computation. However, reaching bigger security parameters of CSIDH seems completely infeasible.[7]

[6] Note that the original formulation was not in terms of orientations, but instead by considering the $\mathbb{F}_p$-rational endomorphisms ring of E. In the language of theorem 2.3.7, the horizontal $\mathbb{Z}[\pi]$-isogenies are precisely the $\mathbb{F}_p$-rational isogenies.

[7] ... at least without a quantum computer, for which the very algorithms that break today's public-key cryptography based on discrete logartihms and factoring, also compute class group structures in polynomial time.

4.1.3 SCALLOP

The notion of more general orientations than the one used in CSIDH was first considered by Colò and Kohel [19]. The case of orientations by orders in $\mathbb{Q}(d\sqrt{-p})$ was studied by Chenu and Smith [15] (see also [14]). The formulation by Colò and Kohel was further built upon by De Feo, Fouotsa, Kutas, Leroux, Merz, Panny, and Wesolowski [31] when they introduced SCALLOP. On a high level, in our formulation, it can simply be seen as a generalisation of CSIDH, where one uses an imaginary quadratic order $\mathfrak{O}$ for which the class group is easy to compute. Specifically, by applying the exact sequence from Theorem 1.2.7, one sees that computing class groups of large conductor reduces to computing the ring of integers (this is the well-known class number formula), and further, one can show that computing the relations in the relation lattice reduces to computing the relations in the ring of integers, and some discrete logarithms.

In practice, this of course leads to lots of changes compared to the CSIDH protocol. Perhaps the biggest problem is that given an ideal $\mathfrak{l}$ of small, prime norm ℓ, the $\mathfrak{l}$-torsion $E[\mathfrak{l}]$ is no longer as easy to compute. First of all, we are back to computing the eigenvectors of the action of δ, where $\mathfrak{O} = \mathbb{Z}[\delta]$, on $E[\ell]$. Further, evaluating the endomorphism δ is also more complicated: In CRS and CSIDH, it was simply Frobenius, while now, δ is not special in any similar way. One way to evaluate δ, as is done in SCALLOP, is to represent δ by a generator of $\ker\delta$; evaluating δ thus comes down to applying Vélu, which again gives smoothness restrictions on $\mathrm{nrd}(\delta)$.

All these requirements makes it particularly tricky to instantiate SCALLOP efficiently. In Paper 6, we study SCALLOP in much greater detail, and come with several improvements, leading to a version of SCALLOP which scales better and is more efficient than the original construction.

4.2 SQIsign

Next, we describe SQIsign, a digital signature scheme, which makes heavy use the computing the Deuring correspondence, the main topic of Part II of this thesis.

Intuitively, SQIsign is based on a Σ-protocol for proving knowledge of the endomorphism ring of a supersingular elliptic curve. Choose a cryptographically sized prime p, and fix a supersingular elliptic curve E_0 defined over $\mathbb{F}_{p^2}$ with known endomorphism ring $\mathcal{O}_0 \subset B_{p,\infty}$. The interactive proof of knowledge proceeds as follows.

Setup: Compute a secret isogeny from $\varphi_A : E_0 \to E_A$. Let $\mathsf{pk} = E_A$, and $\mathsf{sk} = \varphi_A$. Implicitly, φ_A gives knowledge of $\mathrm{End}(E_A) \cong \mathcal{O}_R(I_{\varphi_A})$.

Commit: Compute a random isogeny $\varphi_{\mathsf{commit}} : E_0 \to E_1$, and send the verifier E_1.

Challenge: The verifier computes a random isogeny $\varphi_{\mathsf{chall}} : E_1 \to E_2$, and sends the prover φ_{chall}.

$$\begin{tikzcd}
E_0 \arrow[rr, dotted, "\varphi_{\mathsf{commit}}"] \arrow[dd, dotted, "\varphi_A"'] & & E_1 \arrow[dl, "\varphi'_{\mathsf{chall}}"'] \arrow[dd, "\varphi_{\mathsf{chall}}"] \\
& E_2' & \\
E_A \arrow[ur, "\varphi'_{\mathsf{resp}}"] \arrow[rr, "\varphi_{\mathsf{resp}}"] & & E_2
\end{tikzcd}$$

Figure 4.1: Proof of soundness for SQIsign follows from this diagram. Note that this is *not* a commutative diagram, e.g. $\varphi_{\mathsf{resp}} \circ \varphi_A \neq \varphi_{\mathsf{chall}} \circ \varphi_{\mathsf{commit}}$

Response: The knowledge of $\varphi_{\mathsf{chall}} \circ \varphi_{\mathsf{commit}} : E_0 \to E_2$ gives the prover knowledge of $\mathrm{End}(E_2)$, allowing the prover to compute an isogeny $\varphi_{\mathsf{resp}} : E_A \to E_2$.

Verification: The verifier checks that φ_{resp} is an isogeny from E_A to E_2, such that $\widehat{\varphi_{\mathsf{chall}}} \circ \varphi_{\mathsf{resp}}$ is cyclic.

One can verify that if everyone acts honestly, the verifier accepts, and that if the prover can answer two different challenges $\varphi_{\mathsf{chall}}, \varphi'_{\mathsf{chall}}$ with $\varphi_{\mathsf{resp}}, \varphi'_{\mathsf{resp}}$, then she knows at least one non-integer[8] endomorphism

$$\widehat{\varphi'_{\mathsf{resp}}} \circ \varphi'_{\mathsf{chall}} \circ \widehat{\varphi_{\mathsf{chall}}} \circ \varphi_{\mathsf{resp}} \in \mathrm{End}(E_A).$$

See the diagram in Figure 4.1.

Thus, by answering two different challenges, the prover must be able to compute non-trivial endomorphisms, a problem recently shown to be equivalent to the endomorphism ring problem [59]. Finally, generating transcripts of this protocol is easy, simply by starting with the response isogeny. However, whether such transcripts are indistinguishable from honest transcripts is a very hard question to answer, and depends on many heuristics related to the KLPT algorithm. We refer to the original SQIsign papers for a discussion on this [30, 32].

Going from the high-level description of SQIsign in this section, to an actual efficient implementation, involves a lot of details. In addition to the original papers [30, 32], a thorough description can be found in the specification submitted to the NIST standardisation process [12]. Further, in Paper 3, we study SQIsign in much greater detail and include a very thorough description of the verification procedure.

[8] The fact that this endomorphism is not an integer follows from the cyclicity requirement, and the fact that φ_{chall} and $\varphi'_{\mathsf{chall}}$ are different.

4.3 Other Isogeny-Based Protocols

Perhaps the most important protocols that helped in popularising isogeny-based cryptography were known as Supersingular Isogeny-based Diffie–Hellman (SIDH) [28], and its KEM-variant SIKE [44]. It was also based on a generalization of Diffie–Hellman, where the communicating parties constructed the commutative diagram

$$\begin{tikzcd}
E \arrow[r, "\varphi_A"] \arrow[d, "\varphi_B"'] & E_A \arrow[d, "\varphi'_B"] \\
E_B \arrow[r, "\varphi'_A"'] & E_{AB}
\end{tikzcd}$$

where $\ker \varphi'_A = \varphi_B(\ker \varphi_A)$, and likewise, $\ker \varphi'_B = \varphi_A(\ker \varphi_B)$, making the diagram commute (at least up to isomorphism) by Proposition 2.1.10, arriving at the shared secret $j(E_{AB})$. Clearly, this protocol was reliant on the isogeny problem being hard, however, it also relied on what turned out to be a substantially easier problem.

To make the protocol go through, the parties needed to communicate some extra information, since one party needed to compute $\varphi_B(\ker \varphi_A)$ without the knowledge of φ_B (and vice versa). Writing $N_B := \deg \varphi_A$, this was done by also communicating $\varphi_B(P), \varphi_B(Q)$ for a fixed basis $E[N_B] = \langle P, Q \rangle$. Thus, in the language of Section 2.4, the protocol relied on the isogeny problem being hard for elliptic curves with full N-level structure.

Petit was the first to show that the N-level structure information made the isogeny problem easier, for certain choices of N and degree of the isogeny [61], though even with later improvement by de Quehen, Kutas, Leonardi, Martindale, Panny, Petit, and Stange [34], this was nowhere near being relevant to the parameters used in the SIDH and SIKE protocols. However, this changed drastically in the summer of 2022, when Castryck and Decru [8] used this level structure to completely break SIDH and all its derivatives. This attack was subsequently improved by Maino, Martindale, Panny, Pope, and Wesolowski [55], and finally by Robert [65].

However, at the end of this carnage, we are now beginning to see constructive applications of the insight gained and the powerful techniques developed in these attacks. There now exist many isogeny-based protocols relying on the hardness of more general Γ-level structures. Examples of this include for instance M-SIDH [40] and FESTA [4], though notice that even CSIDH and SCALLOP can be phrased as an isogeny-problem with Γ-level structure since the oriented isogenies fix the eigenspaces of the orientation. For an overview of what is known about isogeny problems with Γ-level structures, we refer to the recent paper by De Feo, Fouotsa, and Panny [33].

Secondly, the core technique from the SIDH-attacks (which were based on what is now known as Kani's lemma [46]), was used by Robert to show that it is possible to evaluate isogenies of any degree in polynomial time [63], as mentioned at the end of Section 2.1. This has already led to a large number of applications in isogeny-based cryptography:

SQIsign-HD uses this technique to instantiate a SQIsign in a new way, which does not require the KLPT algorithm to translate the response ideal to its isogeny. This greatly simplifies the signing procedure, gives shorter signatures, and a much cleaner hardness assumption for the zero-knowledge part, at the cost of a potential slowdown in verification time [27]. In Paper 3, we do the opposite: We complicate the signing procedure in SQIsign, to make verification as fast as possible.

SCALLOP-HD uses this technique to instantiate SCALLOP in a way that makes it easy to instantiate SCALLOP for arbitrarily large security levels, in addition to likely speeding up the group action evaluation [13]. Again, we should compare with Paper 6. Our technique in Paper 6 makes it easier to instantiate higher security levels than SCALLOP, but not as easy as SCALLOP-HD. However, our group action evaluation is the fastest of the three and works only with elliptic curves.

Further results. There has also been found a large number of applications of the techniques, outside of constructing new cryptosystems. We have already mentioned the endomorphism division algorithm [56, Theorem 4.1] (see also [64]), which was important in proving that the problem of computing one endomorphism is equivalent to the endomorphism ring problem [59]. Another example is the paper by Page and Robert [58], which gives a way of evaluating class group actions in polynomial time (recall that the closest we had before this was CSIDH/SCALLOP, which both include a subexponential-time precomputation). Finally, Robert [64] and Kunzweiler and Robert [49] have even applied the techniques to construct new and improved algorithms for the classical problems of computing endomorphism rings of ordinary curves over finite fields, and modular polynomials, respectively.

Thus, what was quickly dubbed, by some, as the end of isogeny-based cryptography, has instead turned out to be the start of a huge and rapid development in the field. No one knows what the future will hold, but one thing about the present is clear: with such a large toolbox, it is an exciting time to do work in isogeny-based cryptography!

Bibliography

[1] Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 411–439. Springer, 2020. doi: 10.1007/978-3-030-64834-3_14. URL https://doi.org/10.1007/978-3-030-64834-3_14.

[2] A Oliver L Atkin and François Morain. Elliptic curves and primality proving. *Mathematics of computation*, 61(203):29–68, 1993.

[3] László Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Comb.*, 6(1):1–13, 1986. URL https://doi.org/10.1007/BF02579403.

[4] Andrea Basso, Luciano Maino, and Giacomo Pope. FESTA: fast encryption from supersingular torsion attacks. In *Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4-8, 2023, Proceedings, Part VII*, volume 14444 of *Lecture Notes in Computer Science*, pages 98–126. Springer, 2023. URL https://doi.org/10.1007/978-981-99-8739-9_4.

[5] Daniel J. Bernstein, Luca De Feo, Antonin Leroux, and Benjamin Smith. Faster computation of isogenies of large prime degree. In Steven Galbraith, editor, *ANTS XIV: Proceedings of the fourteenth algorithmic number theory symposium*. Mathematical Sciences Publishers, 2020. URL https://arxiv.org/abs/2003.10118.

[6] Jean-François Biasse, David Jao, and Anirudh Sankar. A quantum algorithm for computing isogenies between supersingular elliptic curves. In *Progress in Cryptology - INDOCRYPT 2014 - 15th International Conference on Cryptology in India, New Delhi, India, December 14-17, 2014, Proceedings*, volume 8885 of *Lecture Notes in Computer Science*, pages 428–442. Springer, 2014. URL https://doi.org/10.1007/978-3-319-13039-2_25.

[7] Reinier Bröker. Constructing supersingular elliptic curves. *Journal of Combinatorics and Number Theory*, 1(3):269–273, 2009.

[8] Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 423–447. Springer, 2023. URL https://doi.org/10.1007/978-3-031-30589-4_15.

[9] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: an efficient post-quantum commutative group action. In *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part III*, volume 11274 of *Lecture Notes in Computer Science*, pages 395–427. Springer, 2018. URL https://doi.org/10.1007/978-3-030-03332-3_15.

[10] Denis Xavier Charles, Kristin E. Lauter, and Eyal Z. Goren. Cryptographic hash functions from expander graphs. *J. Cryptol.*, 22(1):93–113, 2009. URL https://doi.org/10.1007/s00145-007-9002-x.

[11] Jorge Chávez-Saab, Jesús-Javier Chi-Domínguez, Samuel Jaques, and Francisco Rodríguez-Henríquez. The SQALE of CSIDH: sublinear Vélu quantum-resistant isogeny action with low exponents. *J. Cryptogr. Eng.*, 12(3):349–368, 2022. URL https://doi.org/10.1007/s13389-021-00271-w.

[12] Jorge Chavez-Saab, Maria Corte-Real Santos, Luca De Feo, Jonathan Komada Eriksen, Basil Hess, David Kohel, Antonin Leroux, Patrick Longa, Michael Meyer, Lorenz Panny, Sikhar Patranabis, Christophe Petit, Francisco Rodríguez-Henríquez, Sina Schaeffler, and Benjamin Wesolowski. SQIsign: Algorithm specifications and supporting documentation, 2023. URL https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/spec-files/sqisign-spec-web.pdf. National Institute of Standards and Technology.

[13] Mingjie Chen, Antonin Leroux, and Lorenz Panny. SCALLOP-HD: group action from 2-dimensional isogenies. In *Public-Key Cryptography - PKC 2024 - 27th IACR International Conference on Practice and Theory of Public-Key Cryptography, Sydney, NSW, Australia, April 15-17, 2024, Proceedings, Part III*, volume 14603 of *Lecture Notes in Computer Science*, pages 190–216. Springer, 2024. URL https://doi.org/10.1007/978-3-031-57725-3_7.

[14] Mathilde Chenu. *Supersingular Group Actions and Post-quantum Key Exchange.* PhD thesis, Ecole Polytechnique, 2021.

[15] Mathilde Chenu and Benjamin Smith. Higher-degree supersingular group actions. *IACR Cryptol. ePrint Arch.*, page 955, 2021. URL https://eprint.iacr.org/2021/955.

[16] Andrew M. Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *Journal of Mathematical Cryptology*, 8 (1):1–29, 2014. URL https://doi.org/10.1515/jmc-2012-0016.

[17] Giulio Codogni and Guido Lido. Spectral theory of isogeny graphs. *arXiv preprint arXiv:2308.13913*, 2023.

[18] Henri Cohen. *A course in computational algebraic number theory*, volume 138 of *Graduate texts in mathematics.* Springer, 1993. ISBN 0387556400. URL https://www.worldcat.org/oclc/27810276.

[19] Leonardo Colò and David Kohel. Orienting supersingular isogeny graphs. *Journal of Mathematical Cryptology*, 14(1):414–437, 2020. URL https://doi.org/10.1515/jmc-2019-0034.

[20] J. Brian Conrey, Mark A. Holmstrom, and Tara L. McLaughlin. Smooth neighbors. *Exp. Math.*, 22(2):195–202, 2013. URL https://doi.org/10.1080/10586458.2013.768483.

[21] Giuseppe Cornacchia. Su di un metodo per la risoluzione in numeri interi dell' equazione $\sum_{h=0}^{n} c_h x^{n-h} y^h = p$. *Giornale di Matematiche di Battaglini*, 46:33–90, 1908.

[22] Craig Costello. B-SIDH: Supersingular isogeny diffie-hellman using twisted torsion. In *ASIACRYPT (2)*, volume 12492 of *Lecture Notes in Computer Science*, pages 440–463. Springer, 2020. URL https://ia.cr/2019/1145.

[23] Craig Costello, Michael Meyer, and Michael Naehrig. Sieving for Twin Smooth Integers with Solutions to the Prouhet-Tarry-Escott Problem. In *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part I*, volume 12696 of *Lecture Notes in Computer Science*, pages 272–301. Springer, 2021. URL https://doi.org/10.1007/978-3-030-77870-5_10.

[24] Jean Marc Couveignes. Hard homogeneous spaces. *IACR Cryptol. ePrint Arch.*, 2006. URL http://eprint.iacr.org/2006/291.

[25] Jean-Marc Couveignes. Quelques math\'ematiques de la cryptographie\a cl\'es publiques. *Journée annuelle de la SMF*, 2007.

[26] David A. Cox. *Primes of the Form $x^2 + ny^2$: Fermat, Class Field Theory, and Complex Multiplication*. Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts. Wiley, 2013.

[27] Pierrick Dartois, Antonin Leroux, Damien Robert, and Benjamin Wesolowski. SQIsignHD: New dimensions in cryptography. In *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part I*, volume 14651 of *Lecture Notes in Computer Science*, pages 3–32. Springer, 2024. URL https://doi.org/10.1007/978-3-031-58716-0_1.

[28] Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Journal of Mathematical Cryptology*, 8 (3):209–247, 2014. URL https://ia.cr/2011/506.

[29] Luca De Feo, Jean Kieffer, and Benjamin Smith. Towards practical key exchange from ordinary isogeny graphs. In *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part III*, volume 11274 of *Lecture Notes in Computer Science*, pages 365–394. Springer, 2018. URL https://doi.org/10.1007/978-3-030-03332-3_14.

[30] Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. SQISign: Compact post-quantum signatures from quaternions and isogenies. In *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 64–93. Springer, 2020. URL https://doi.org/10.1007/978-3-030-64837-4_3.

[31] Luca De Feo, Tako Boris Fouotsa, Péter Kutas, Antonin Leroux, Simon-Philipp Merz, Lorenz Panny, and Benjamin Wesolowski. SCALLOP: scaling the CSI-FiSh. In *Public-Key Cryptography - PKC 2023 - 26th IACR International Conference on Practice and Theory of Public-Key Cryptography, Atlanta, GA, USA, May 7-10, 2023, Proceedings, Part I*, volume 13940 of *Lecture Notes in Computer Science*, pages 345–375. Springer, 2023. URL https://doi.org/10.1007/978-3-031-31368-4_13.

[32] Luca De Feo, Antonin Leroux, Patrick Longa, and Benjamin Wesolowski. New algorithms for the deuring correspondence - towards practical and secure SQISign signatures. In *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*, volume 14008 of

Lecture Notes in Computer Science, pages 659–690. Springer, 2023. URL https://doi.org/10.1007/978-3-031-30589-4_23.

[33] Luca De Feo, Tako Boris Fouotsa, and Lorenz Panny. Isogeny problems with level structure. *IACR Cryptol. ePrint Arch.*, 2024. URL https://eprint.iacr.org/2024/459.

[34] Victoria de Quehen, Péter Kutas, Chris Leonardi, Chloe Martindale, Lorenz Panny, Christophe Petit, and Katherine E. Stange. Improved torsion-point attacks on SIDH variants. In *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part III*, volume 12827 of *Lecture Notes in Computer Science*, pages 432–470. Springer, 2021. URL https://doi.org/10.1007/978-3-030-84252-9_15.

[35] Christina Delfs and Steven D. Galbraith. Computing isogenies between supersingular elliptic curves over $\mho_p$. *Des. Codes Cryptogr.*, 78(2):425–440, 2016. URL https://doi.org/10.1007/s10623-014-0010-1.

[36] Max Deuring. Die Typen der Multiplikatorenringe elliptischer Funktionenkörper. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 14:197–272, 1941.

[37] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Trans. Inf. Theory*, 22(6):644–654, 1976. URL https://doi.org/10.1109/TIT.1976.1055638.

[38] Kirsten Eisenträger, Sean Hallgren, Chris Leonardi, Travis Morrison, and Jennifer Park. Computing endomorphism rings of supersingular elliptic curves and connections to path-finding in isogeny graphs. *Open Book Series*, 4(1):215–232, 2020.

[39] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986. URL https://doi.org/10.1007/3-540-47721-7_12.

[40] Tako Boris Fouotsa, Tomoki Moriya, and Christophe Petit. M-SIDH and MD-SIDH: countering SIDH attacks by masking information. In *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 282–309. Springer, 2023. URL https://doi.org/10.1007/978-3-031-30589-4_10.

[41] Jenny Fuselier, Annamaria Iezzi, Mark Kozek, Travis Morrison, and Changningphaabi Namoijam. Computing supersingular endomorphism rings using inseparable endomorphisms. *arXiv preprint arXiv:2306.03051*, 2023.

[42] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inf. Theory*, 31(4):469–472, 1985. doi: 10.1109/TIT.1985.1057074. URL https://doi.org/10.1109/TIT.1985.1057074.

[43] James L Hafner and Kevin S McCurley. A rigorous subexponential algorithm for computation of class groups. *Journal of the American mathematical society*, 2(4):837–850, 1989.

[44] David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, Vladimir Soukharev, David Urbanik, Geovandro Pereira, Koray Karabina, and Aaron Hutchinson. SIKE, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions.

[45] 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. *Mathematical Cryptology*, 1(2):10–24, 2022. URL https://journals.flvc.org/mathcryptology/article/view/130618.

[46] Ernst Kani. The number of curves of genus two with elliptic differentials. *Journal für die reine und angewandte Mathematik*, pages 93–122, 1997.

[47] 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.

[48] David Kohel, Kristin E. Lauter, Christophe Petit, and Jean-Pierre Tignol. On the quaternion ℓ-isogeny path problem. In *ANTS XI: Proceedings of the eleventh algorithmic number theory symposium*. Mathematical Sciences Publishers, 2024.

[49] Sabrina Kunzweiler and Damien Robert. Computing modular polynomials by deformation. In *ANTS XVI: Proceedings of the sixteenth algorithmic number theory symposium*. Mathematical Sciences Publishers, 2024.

[50] Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM J. Comput.*, 35(1):170–188, 2005. URL https://doi.org/10.1137/S0097539703436345.

[51] Tom Leinster. *Basic category theory*, volume 143 of *Cambridge studies in advanced mathematics*. Cambridge University Press, 2014.

[52] Arjen K Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische annalen*, 261:515–534, 1982.

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

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

[55] Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 448–471. Springer, 2023. URL https://doi.org/10.1007/978-3-031-30589-4_16.

[56] Arthur Herlédan Le Merdy and Benjamin Wesolowski. The supersingular endomorphism ring problem given one endomorphism. *IACR Cryptol. ePrint Arch.*, 2023. URL https://eprint.iacr.org/2023/1448.

[57] Hiroshi Onuki. On oriented supersingular elliptic curves. *Finite Fields Their Appl.*, 69:101777, 2021. URL https://doi.org/10.1016/j.ffa.2020.101777.

[58] Aurel Page and Damien Robert. Introducing Clapoti(s): Evaluating the isogeny class group action in polynomial time. *IACR Cryptol. ePrint Arch.*, 2023. URL https://eprint.iacr.org/2023/1766.

[59] Aurel Page and Benjamin Wesolowski. The supersingular endomorphism ring and one endomorphism problems are equivalent. In *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part VI*, volume 14656 of *Lecture Notes in Computer Science*, pages 388–417. Springer, 2024. URL https://doi.org/10.1007/978-3-031-58751-1_14.

[60] Trevor Perrin and Moxie Marlinspike. The double ratchet algorithm. *GitHub wiki*, 112, 2016.

[61] Christophe Petit. Faster algorithms for isogeny problems using torsion point images. In *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*, volume 10625 of *Lecture Notes in Computer Science*, pages 330–353. Springer, 2017. URL https://doi.org/10.1007/978-3-319-70697-9_12.

[62] Dimitrij Ray. Constructing the Deuring correspondence with applications to supersingular isogeny-based cryptography. Master's thesis, Technische Universiteit Eindhoven, 2018.

[63] Damien Robert. Evaluating isogenies in polylogarithmic time. *IACR Cryptol. ePrint Arch.*, 2022. URL `https://eprint.iacr.org/2022/1068`.

[64] Damien Robert. Some applications of higher dimensional isogenies to elliptic curves (preliminary version). *IACR Cryptol. ePrint Arch.*, 2022. URL `https://eprint.iacr.org/2022/1704`.

[65] Damien Robert. Breaking SIDH in polynomial time. In *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 472–503. Springer, 2023. URL `https://doi.org/10.1007/978-3-031-30589-4_17`.

[66] Alexander Rostovtsev and Anton Stolbunov. Public-key cryptosystem based on isogenies. *IACR Cryptol. ePrint Arch.*, 2006. URL `http://eprint.iacr.org/2006/145`.

[67] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252. Springer, 1989. URL `https://doi.org/10.1007/0-387-34805-0_22`.

[68] René Schoof. Elliptic curves over finite fields and the computation of square roots mod p. *Mathematics of computation*, 44(170):483–494, 1985.

[69] René Schoof. Counting points on elliptic curves over finite fields. *Journal de théorie des nombres de Bordeaux*, 7(1):219–254, 1995.

[70] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 124–134. IEEE Computer Society, 1994. URL `https://doi.org/10.1109/SFCS.1994.365700`.

[71] Joseph H. Silverman. *The Arithmetic of Elliptic Curves*, volume 106 of *Graduate Texts in Mathematics*. Springer, 2 edition, 2009. ISBN 978-0-387-09493-9.

[72] Anton Stolbunov. Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves. *Adv. Math. Commun.*, 4(2):215–235, 2010. URL `https://doi.org/10.3934/amc.2010.4.215`.

[73] Anton Stolbunov. *Cryptographic Schemes Based on Isogenies*. PhD thesis, Norwegian University of Science and Technology, 2012.

[74] Andrew V Sutherland. *Order computations in generic groups.* PhD thesis, Massachusetts Institute of Technology, 2007.

[75] John Tate. Endomorphisms of abelian varieties over finite fields. *Inventiones mathematicae*, 2(2):134–144, 1966.

[76] Luke Valenta, Nick Sullivan, Antonio Sanso, and Nadia Heninger. In search of CurveSwap: Measuring elliptic curve implementations in the wild. In *2018 IEEE European Symposium on Security and Privacy, EuroS&P 2018, London, United Kingdom, April 24-26, 2018*, pages 384–398. IEEE, 2018. URL https://doi.org/10.1109/EuroSP.2018.00034.

[77] Jacques Vélu. Isogénies entre courbes elliptiques. *Comptes Rendus de l'Académie des Sciences de Paris*, 273(4):238–241, 1971. URL https://gallica.bnf.fr/ark:/12148/bpt6k56191248/f52.item.

[78] John Voight. *Quaternion Algebras*, volume 288. Springer Graduate Texts in Mathematics series, 2018.

[79] William C. Waterhouse. Abelian varieties over finite fields. *Annales scientifiques de l'École Normale Supérieure*, 2:521–560, 1969.

[80] Benjamin Wesolowski. The supersingular isogeny path and endomorphism ring problems are equivalent. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 1100–1111. IEEE, 2021. URL https://doi.org/10.1109/FOCS52979.2021.00109.

[81] Benjamin Wesolowski. Orientations and the supersingular endomorphism ring problem. In *EUROCRYPT (3)*, volume 13277 of *Lecture Notes in Computer Science*, pages 345–371. Springer, 2022. URL https://ia.cr/2021/1583.

Part 2

The Constructive Deuring Correspondence

There is this theorem, you see,
where orders and curves are the key;
They're easy to relate,
and share many 'a trait,
e.g. both are incomprehensible to me!

Jonathan Komada Eriksen,
February 2023

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

Jonathan Komada Eriksen, Lorenz Panny, Jana Sotáková, and Mattia Veroni

Accepted at LuCaNT 2023. Full version available at:
eprint.iacr.org/2023/106.pdf

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 Veron[1]

[1] Norwegian University of Science and Technology, Trondheim, Norway
[2] Academia Sinica, Taipei, Taiwan
[3] University of Amsterdam and QuSoft, Amsterdam, The Netherlands

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

Author list in alphabetical order; see `https://ams.org/profession/leaders/CultureStatement04.pdf`. This research was funded in part by the Academia Sinica Investigator Award AS-IA-109-M01 and the Dutch Research Council (NWO) through Gravitation-grant Quantum Software Consortium – 024.003.037.

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 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 $p \not\equiv 3 \pmod 4$.

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 current de-facto 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.

Acknowledgements. Many thanks to the anonymous reviewers and to John Cremona and Andrew Sutherland for their insightful comments, which (among other things) led to a neater proof of Lemma 5.

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 say that N is *power-smooth* if for any prime factor $q \mid N$, the largest power of q dividing N is smaller than B 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].

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

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 non-constant 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 gives rise to 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 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 (geometric) *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* $\pi\colon\ E \to E,$, given by $(x, y) \mapsto (x^q, y^q)$. 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, that is, $\pi_{\tilde{E}} = -\pi_E$.

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, and let $k \in \mathbb{Z}_{>0}$. Then the p-power Frobenius π of E satisfies $\pi^2 = -p$ and*

$$\#E(\mathbb{F}_{p^{2k}}) = \big(p^k - (-1)^k\big)^2.$$

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 1. *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 1. In Theorem 1, 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 1 that $\pi = -p$ can (say, in algorithms) always be satisfied by taking an isomorphism:

Lemma 2. *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 we are 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 $\pi^2 - mp\pi + p^2 = 0$ 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 model 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 2 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. It also explains why we do not make the distinction between the geometric endomorphism ring and rational endomorphism ring: all the endomorphisms are already defined over $\mathbb{F}_{p^2}$.

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 cyclic, and one is given a generator 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 $\sqrt{\text{é}}$lu. 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 already uniquely defined by any irreducible divisor of the kernel polynomial; we exploit this in Algorithm 3. An algorithm which interpolates between the $\sqrt{\text{é}}$lu and Kohel approaches by working with irreducible (rational!) divisors of the kernel polynomial is outlined in [BDLS20, § 4.14], where it is argued that this algorithm cannot be expected to improve upon Kohel's formulas 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:

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

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. A quaternion algebra is determined up to isomorphism by the set of its ramified primes. 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 a maximal 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 $\operatorname{nrd}(I) = \gcd(\{\operatorname{nrd}(\alpha) : \alpha \in I\})$; 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 $I = \mathcal{O}_L(I)\alpha + \mathcal{O}_L(I)\operatorname{nrd}(I)$ for any $\alpha \in \mathcal{O}_L(I)$ satisfying $\gcd(\operatorname{nrd}(\alpha), \operatorname{nrd}(I)^2) = \operatorname{nrd}(I)$, 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 $\operatorname{nrd}(IJ) = \operatorname{nrd}(I)\operatorname{nrd}(J)$.

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 $\operatorname{nrd}(\omega_1) = q$ and $\operatorname{nrd}(\omega_2) = p$ 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 of supersingular elliptic curves over $\overline{\mathbb{F}}_p$ under isogenies, denoted as SS_p. As for any supersingular elliptic curve E, its endomorphism ring $\operatorname{End}(E)$ is isomorphic to a maximal order in the quaternion algebra $B_{p,\infty}$. We will see that the homsets $\operatorname{Hom}(E, E')$ between two elliptic curves also carry extra structure.

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

$$\operatorname{Hom}(E, E') \times \mathcal{O} \to \operatorname{Hom}(E, E')$$
$$(\phi, \alpha) \mapsto \phi \circ \rho(\alpha)$$

turns $\operatorname{Hom}(E, E')$ into a right $\mathcal{O}$-module. With some extra work, one can show that $\operatorname{Hom}(E, E')$ is not only a right $\mathcal{O}$-module, but in fact isomorphic to a right $\mathcal{O}$-ideal. Completely analogously, $\operatorname{Hom}(E, E')$ 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 $\operatorname{lfrac} \mathcal{O}_0$.

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

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

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 $\operatorname{End}(E) \cong \mathcal{O}$. 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 $p \nmid \operatorname{nrd}(I)$. 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 $\operatorname{Hom}(E_I, E_0) \cong I$ (where the isomorphism is given by sending $\psi \in \operatorname{Hom}(E_I, E_0)$ to $\psi \circ \phi_I \in I$), 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 $\deg(\phi_I) = \operatorname{nrd}(I)$ 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 3.

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 1 (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 $\operatorname{End}(E) \cong \mathcal{O}$.

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 $\alpha \in \operatorname{End}(E)$ 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 2 (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 $\operatorname{End}(E)$ as a $\mathbb{Z}$-lattice.

Effective endomorphisms. Answers to Problems 1 and 2 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 $\phi_j \in \operatorname{End}(E)$ in efficient representation, such that mapping $\alpha_j \mapsto \phi_j$ defines a ring isomorphism between $\mathcal{O}$ and $\operatorname{End}(E)$. 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 [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 $\operatorname{Hom}(E, E') \otimes_{\mathbb{Z}} \mathbb{Q}$. 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, one can 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 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*

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

basis[5] 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 2. 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 3 (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 Deuring is the MaxOrder problem (Problem 1).

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 2, 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.

[5] For coprime $(m_1, ..., m_r)$, a list of integers $(a_1, \ldots, a_r)$ such that $a_i \equiv 1 \pmod{m_i}$ $\forall i$ and $a_i \equiv 0 \pmod{m_j}$ $\forall i \neq j$.

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 $\operatorname{End}(E) \cong \mathcal{O}$.

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 which involves 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 the technique from 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, for ζ_3 a non-trivial cube root of unity. 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 remains inert in $\mathbb{Q}(\sqrt{-q})$, compute the unique (see [CX22, Theorem 1.1] and [Cox22, Proposition 3.11]) rational 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 3. 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 2.

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 3. *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 $\gamma := \mathbf{i}^{-1}f(\mathbf{i}')$ is of the form $x + y\mathbf{j}$ with $x, y \in \mathbb{Q}$, and we have $x^2 + py^2 = \mathrm{nrd}(\gamma) = q'/q$ by multiplicativity of the norm.

Correctness of the constructed isomorphism is readily verified.

The sledgehammer method to find the pair (x, y) in Lemma 3 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

$$I(\mathcal{O}_0, \mathcal{O}) := N\mathcal{O}_0\mathcal{O}. \tag{2}$$

It is easy to see that it is a connecting ideal, and it is clearly integral. Note also that N is minimal possible such that $I(\mathcal{O}_0, \mathcal{O})$ is integral, as can be seen from requiring $N\mathcal{O} \subset N\mathcal{O}_0\mathcal{O} \subset \mathcal{O}_0$. 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 4 ([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 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 iterating 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 4 (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.

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.

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

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 4) 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 1 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 J-torsion $E_0[J]$. Algorithms for doing this were first presented by Galbraith, Petit and Silva [GPS17]. Our version is based on this but includes 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 generate the bases of the torsion subgroups $E_0[\ell_i^{e_i}]$.

Generating bases of torsion groups. We need to generate a basis for the torsion groups $E_0[\ell_i^{e_i}]$ for all $\ell_i^{e_i} \mid \mathrm{nrd}(J)$. Let k be an integer such $E_0[\ell_i^{e_i}] \subseteq E_0(\mathbb{F}_{p^{2k}})$. 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 1. 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 $E_0[\ell_i^{e_i}]$, we also need to check linear independence (for instance, by checking that the weil pairing $e_{\ell_i^{e_i}}(P, Q)$ is a primitive $\ell_i^{e_i}$-root of unity). 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.

So generating the torsion bases costs $O(k \log p) \cdot M(k)$, for $M(k)$ the cost of multiplying in $\mathbb{F}_{p^{2k}}$.

Finding kernel generators. Once we have the bases for $E_0[\ell_i^{e_i}]$, we can compute $E_i[J]$. Following [GPS17], the idea is to find the action on the torsion subgroup $E_0[\ell_i^{e_i}]$ 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 $\langle P, Q \rangle = E_0[\ell_i^{e_i}]$ by means of the techniques discussed in Section 2.6. By computing two discrete logarithms (easy if ℓ_i 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 $\widetilde{O}(\ell_i M(k_i))$. (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 $\sqrt{\text{é}}\text{lu}$ 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 5 (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 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 $\mathrm{nrd}(J) = N$.

We will assume that J is cyclic, that is, it is does not factor as $J = mK$ for any integer $m \neq \pm 1$ and integral ideal K; otherwise we scale J by a suitable scalar. (Equivalently, the kernel of the corresponding isogeny is a cyclic subgroup.) In this case, 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 6. 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 Section 3.1). 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 $\ell_i$-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.

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 4) $B = p^3$. 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 $E[\ell^e]$ 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 $\mathbb{F}_{p^{2k}}$;
- 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, and may be far from optimal. First, it follows the asymptotic costs of the relevant algorithms, which might be significantly different for practical values of p. Then, the values c_i, $i = 1, \ldots, 4$ have been computed by optimising the average runtime by trial-and-error, strongly depending on the cost model. Fine-tuning this torsion-optimization step, that is improving the cost-model and then recomputing the constant, can almost certainly lead to better results.

A picture is worth a thousand words. To illustrate why our approach can lead to improvements, we examine the following two figures. In Figure 1, for each prime power ℓ^e ($\ell \geq 2, e \geq 1$) between 2 and 101 on the x-axis, we plot on the y-axis the extension degrees — different primes may require different extension degrees, hence the plural — of $\mathbb{F}_{p^2}$ required to define the full ℓ^e-torsion subgroup . The intensity of the color of the pixels corresponds to the probability of obtaining degree k when p varies. 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 so usually ends up with extension degrees depicted in the upper part of Figure 1 (as those are the most probable ones).

Our approach corresponds to looking at a "wider" picture for each particular p: in Figure 2, each data point still corresponds to the extension degree over $\mathbb{F}_{p^2}$ over which the ℓ^e-torsion is defined, but we allow for larger values of ℓ^e. We then choose data points in the lower part of the picture, speeding up computations by skipping values ℓ^e which need large extension degrees k, when working with these would require — based on the cost model — more computations than using larger values ℓ^e with smaller extension degrees k.

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 computes with torsion which does not require large extension degrees.

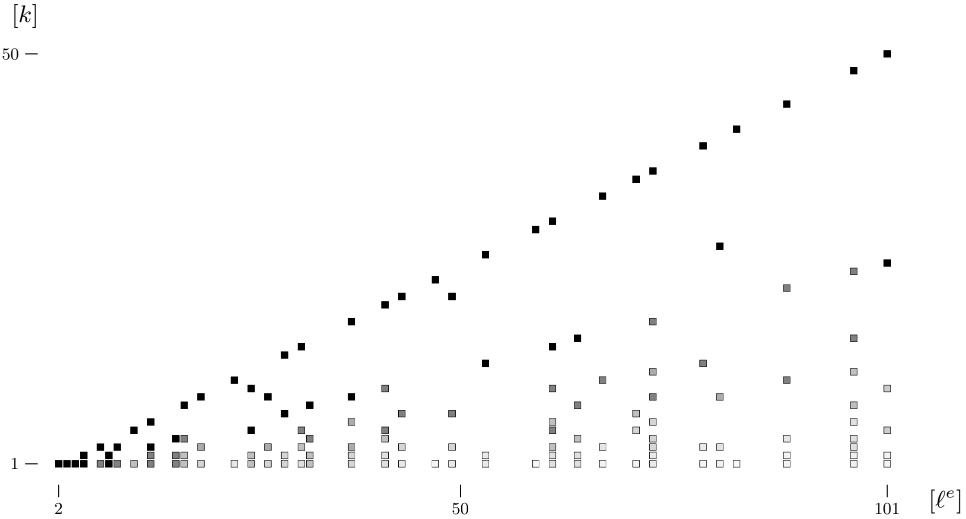


Fig. 1. Heatmap of distribution of extension degrees k required to access the ℓ^e-torsion subgroup of a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$, under the heuristic assumption that p behaves like a random unit modulo all ℓ^e. The intensity with which each data point (ℓ^e, k) is drawn 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).

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

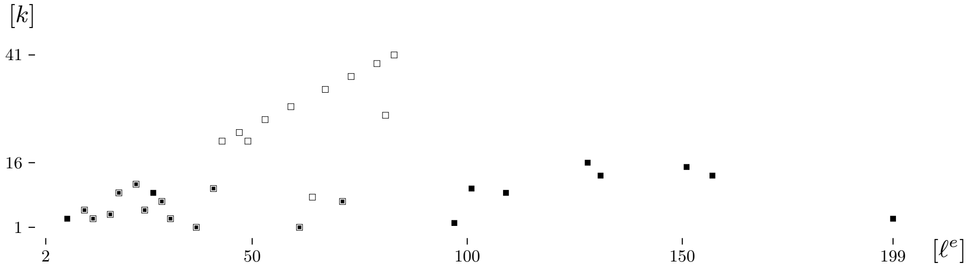


Fig. 2. Illustration of ℓ^e-torsion subgroups with their associated extension degree k chosen by naïve powersmooth KLPT (□) and our algorithm (■), for a quadratic-time cost model and some particular combination of prime p and magnitude of norm of the output ideal.

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 1, 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 done 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 $\mathbb{F}_q$-algebra.

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 1. *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 H 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 H, and if f is furthermore irreducible, we refer to f as a minimal polynomial *of H.*

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).

An immediate application is computing isogenies from irrational kernel points:

Lemma 5. *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 $\ell = 2$, then $k = 1$ and $m = 1$ and the algorithm simply returns f. Assume $\ell \geq 3$ below.

Fix any point $P \in E\backslash\{0\}$ with $f(x(P)) = 0$. As ℓ is prime, P is a generator of H. Let π denote the q-power Frobenius endomorphism of E. Since $x(\pi(P)) = x(P)^q$ is a Galois conjugate of a root of f, it must itself be a root of f, and therefore $\pi(P)$ also generates H. Hence, there exists a scalar $\lambda \in (\mathbb{Z}/\ell)^\times$ such that $\pi(P) = [\lambda]P$, and since all other points in H are scalar multiples of P we conclude that H is a λ-eigenspace of Frobenius. In other words, the Galois action on $H\backslash\{0\}$ factors through the (free and transitive) action of $(\mathbb{Z}/\ell)^\times$ on the set $H\backslash\{0\}$ of generators of H: Its image is $\langle\lambda\rangle \leq (\mathbb{Z}/\ell)^\times$.

We quotient everything by negation, yielding a (still free and transitive) action of $A := (\mathbb{Z}/\ell)^\times/\pm$ on the set $X := (H\backslash\{0\})/\pm = \{\{Q, -Q\} : Q \in H\backslash\{0\}\}$. Via the

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

```
Input: Elliptic curve $E/\mathbb{F}_q$, prime integer $\ell$, minimal polynomial $f \in \mathbb{F}_q[X]$ of an order-$\ell$
       subgroup $H \leq E$.
Output: The kernel polynomial $h \in \mathbb{F}_q[X]$ of $H$.
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 $\ell$ 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 $\alpha_i$ 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-$\ell$
       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 $\xi$ over $\mathbb{F}_q$ using Shoup's algorithm.
Return KernelPolynomialFromDivisor$(E, \mu, \ell)$.
```

x-coordinate projection, the latter is in bijection with the set $\{x(Q) : Q \in H\backslash\{0\}\}$ of roots of the desired kernel polynomial, and this correspondence is compatible with the respective Galois actions. Thus, the irreducible factors of the desired kernel polynomial correspond to Galois orbits of the subgroup $S := \langle\lambda\rangle/\pm$ on the A-set X. Since the orbit corresponding to f has size k, all other orbits must have the same size. The orbits are in bijection with the quotient group A/S, and they can be enumerated by acting with any transversal of A/S: The algorithm uses the first m powers of a.

Concretely, for $i \geq 2$, the algorithm computes the polynomial f_i corresponding to the i-th Galois orbit as the minimal polynomial over $\mathbb{F}_q$ of an explicitly constructed root α_i of f_i. Indeed, throughout, the quotient ring $\mathbb{F}_q[X]/f_{i-1}$ is a representation of the finite field $\mathbb{F}_{q^k}$, and the images of the α_i in $\mathbb{F}_{q^k}$ form a set of representatives for the Galois orbits of the roots of the kernel polynomial.

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 7. 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 8. 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 H, 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 1). The technique generalizes Steps 4 and 5 of Algorithm 3.

Lemma 6. *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 that $P \in \ker\varphi$. Otherwise, we have $f_1(\xi) = 0$ and hence $\iota_1(\alpha) = g_1(\xi)/g_2(\xi) = x(\varphi(P))$ 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 $x(\varphi(P))$. However, since f^φ was defined independently of the

Algorithm 5: `PushSubgroup`(E, f, φ)

```
Input: Elliptic curve $E/\mathbb{F}_q$, minimal polynomial $f \in \mathbb{F}_q[X]$ of a subgroup $H \leq E$,
       isogeny $\varphi\colon E \to E'$ defined over $\mathbb{F}_q$.
Output: Minimal polynomial $f^\varphi \in \mathbb{F}_q[X]$ of the subgroup $\varphi(H) \leq E'$.
Write the x-coordinate map of $\varphi$ as a fraction $g_1/g_2$ of polynomials $g_1, g_2 \in \mathbb{F}_q[X]$.
Let $g_{\text{ker}} \leftarrow \gcd(g_2, f)$ and $f_1 \leftarrow f/g_{\text{ker}}$.
Compute $g_1 \cdot g_2^{-1} \bmod f_1 \in \mathbb{F}_q[X]$ and reinterpret it as a quotient-ring element
    $\alpha \in \mathbb{F}_q[X]/f_1$.
Find the minimal polynomial $f^\varphi \in \mathbb{F}_q[X]$ of $\alpha$ over $\mathbb{F}_q$ using Shoup's algorithm.
Return $f^\varphi$.
```

particular P considered, we see that f^φ in fact vanishes at $x(\varphi(P))$ for *all* such points P. Thus, f^φ is indeed a defining polynomial for the group $\varphi(H)$ generated by all the $\varphi(P)$, as claimed.

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 H with coordinates in $\mathbb{F}_{q^k}$ uses $O(nk(\log k)^{O(1)})$ operations in $\mathbb{F}_q$.

Remark 9. 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 examples illustrating the tools developed in Section 4: in Example 1 we give a worked example of the entire computation for a prime $p \equiv 1 \pmod{12}$, and in Example 2 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 1. 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 1)

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

In order to find a curve with endomorphism ring isomorphic to $\mathcal{O}$, we need a connecting $(\mathcal{O}_0, \mathcal{O})$-ideal. We make the same choice as in Equation (2):

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

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

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

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

Step 2. Next, we find generators for the full torsion groups (recall, we need an extra power of 2 because of the denominators):

$$E_0[2^8] \subseteq E_0(\mathbb{F}_{p^4}), \quad E_0[3^4] \subseteq \widetilde{E_0}(\mathbb{F}_{p^{18}}), \quad E_0[5^3] \subseteq \widetilde{E_0}(\mathbb{F}_{p^4}), \quad E_0[11] \subseteq \widetilde{E_0}(\mathbb{F}_{p^5}), \quad E_0[13] \subseteq \widetilde{E_0}(\mathbb{F}_{p^6}),$$

$$E_0[17] \subseteq \widetilde{E_0}(\mathbb{F}_{p^{16}}), \quad E_0[19] \subseteq E_0(\mathbb{F}_{p^{18}}), \quad E_0[29] \subseteq \widetilde{E_0}(\mathbb{F}_{p^4}), \quad E_0[53] \subseteq \widetilde{E_0}(\mathbb{F}_{p^2}).$$

Here, $\widetilde{E}(\mathbb{F}_q)$ denotes the group of $\mathbb{F}_q$-rational points on a quadratic twist of E over $\mathbb{F}_q$.

Next, we determine the action of J on the different torsion groups (see Algorithm 2), and find a generator P_{ℓ^e} for every (maximal) prime power $\ell^e \mid \mathrm{nrd}(J)$. 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

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

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

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

Example 2 (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_0 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_1 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 $B = (-1, -p)$, in order to recover the embedding $\mathcal{O}_1' \hookrightarrow \mathcal{O}_0 \otimes \mathbb{Q}$ from Remark 2. To apply Lemma 3, 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 3 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 $I = N\mathcal{O}_0\mathcal{O}_1$, with N as in Equation (2), 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 1, 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 group of $\mathbb{F}_q$-rational points on 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 1). 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

$$\begin{tikzcd} E_0 \arrow[r, "\varphi_3"] & E' \arrow[r, "\varphi_{17}"] & E'' \arrow[r, "\varphi_{97}"] & E_1 \end{tikzcd}$$

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 computing 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. We run the computation on three different primes of ≈ 256 bits:

- p_{3923}: The 254-bit prime currently used in SQISign [DLW22],

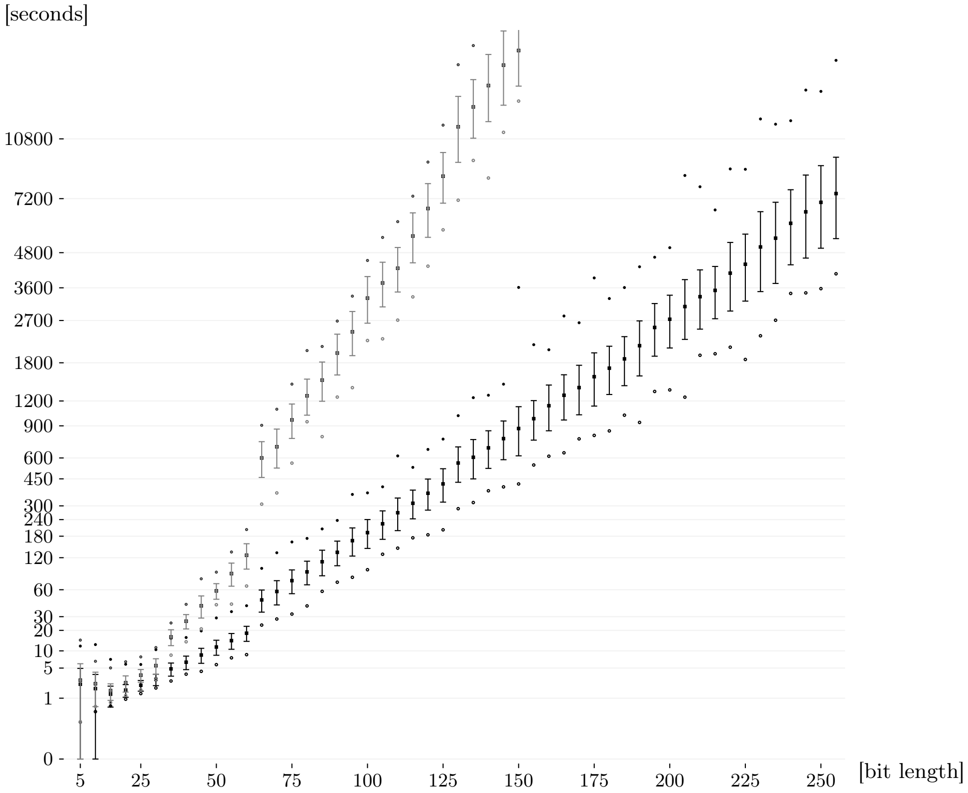


Fig. 3. Timings for running our implementation of the Deuring correspondence for random primes up to 255 bits (in steps of 5 bits), showing mean (■), minimum (○), maximum (•), and estimated standard deviation (error bars). The y-axis uses a quartic scale. Each data point represents measurements from 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.

- 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).

We also see that our algorithm runs fastest for p_1. While p_2 and p_{3923} were 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 10 (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 three 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. The third column lists the torsion subgroups that required the top 3 longest runtimes, together with the required extension degree.

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*. Preprint. 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.

[Cox22] David A Cox. *Primes of the Form x2+ ny2: Fermat, Class Field Theory, and Complex Multiplication. with Solutions*. Vol. 387. American Mathematical Soc., 2022.

[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 Noninteger 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:

$$\begin{tikzcd}
E_0 \arrow[r, "\phi_1"] \arrow[rrrrr, bend right, "\phi_I"'] & E_1 \arrow[r, "\phi_2"] & E_2 \arrow[r, "\phi_3"] & \dots \arrow[r, "\phi_{f-1}"] & E_{f-1} \arrow[r, "\phi_f"] & E_f
\end{tikzcd}$$

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 2.

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 $k = 1, \ldots, f$ then translates the whole ideal I. This "sliding" procedure is summarized in Figure 4.

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''

$$\begin{tikzcd}
E_0 \arrow[r, "\phi_1"] \arrow[r, dashed, bend right, "\psi_1"'] \arrow[rr, dashed, bend right, "\psi_2"'] \arrow[rrr, dashed, bend right, "\cdots"'] \arrow[rrrr, dashed, bend right, "\psi_{f-1}"'] & E_1 \arrow[r, "\phi_2"] & E_2 \arrow[r, "\phi_3"] & \dots \arrow[r, "\phi_{f-1}"] & E_{f-1} \arrow[r, "\phi_f"] & E_f
\end{tikzcd}$$

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

by computing the kernel of the pullback ideal $[I]^*\overline{J''}$, and push it through ϕ_I to obtain $\ker \phi_{\bar{J}''} = \ker \widehat{\phi_{J''}}$. One can then recover ϕ_J as $\widehat{\phi_{\bar{J}''}} \circ \phi_{J'}$.

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: $\phi_I$.
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 $J_i I_i + \mathcal{O}_0 \mathrm{nrd}(I_i)$.
    Compute $K$ as the group generated by IdealToKernelGens$([J_i]^* I_i, E_0)$.
    Compute $\phi_{I_i}$ 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 $\phi_{I_f} \circ \cdots \circ \phi_{I_1}$.
```

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 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 performs worse for us in practice. We do not rule out the possibility that this technique may become faster with different parameters and optimizations.

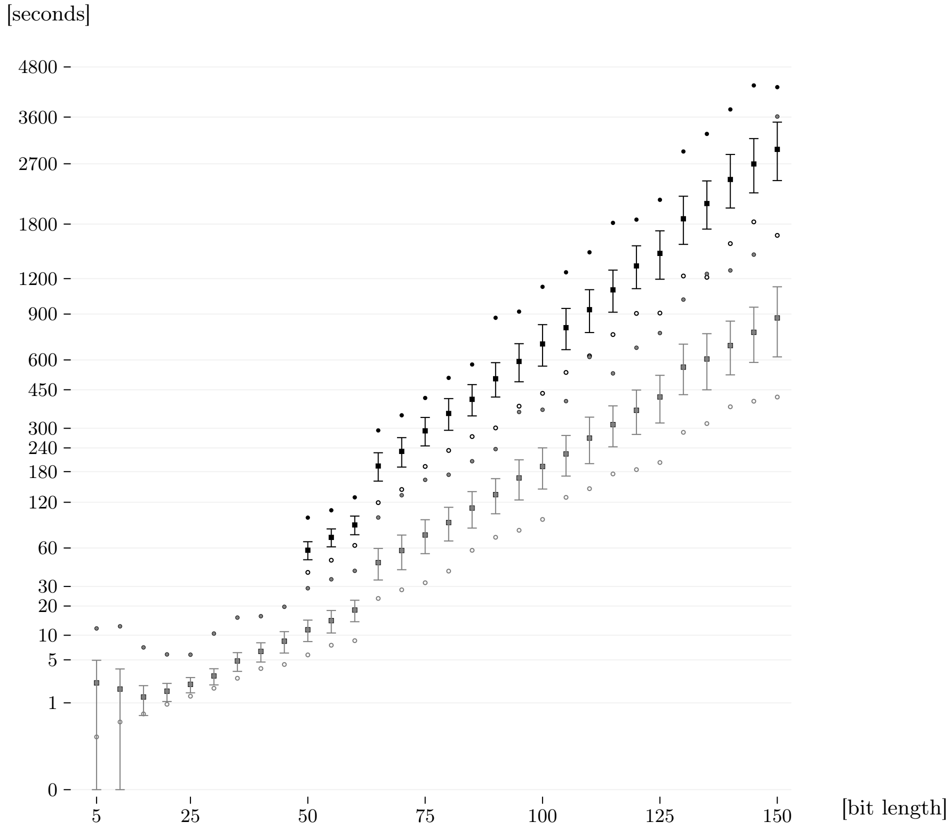


Fig. 5. Timings for running our implementation of the Deuring correspondence using Algorithm 6 for primes up to 150 bits, with the same experimental setup as for Figure 3. Each data point represents measurements from 128 independent runs. Data points from Figure 3 shown in gray for reference.

B Custom primes used in experiments

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

$$\begin{aligned}
p_1 &= 11956566944641502957704189594909498993478297403838643406058180334130656750161 \\
p_2 &= 37670568336551536389503919665937491111216122470333837677213877442445311999999 \\
p_{3923} &= 23759399264157352358673788613307970528646815114090876784643387662192449945599
\end{aligned}$$

Cryptographic Smooth Neighbors

Giacomo Bruno, Maria Corte-Real Santos, Craig Costello, Jonathan Komada Eriksen, Michael Meyer, Michael Naehrig, and Bruno Sterner

Accepted at AsiaCrypt 2023. Full version available at:
eprint.iacr.org/2022/1439.pdf

Cryptographic Smooth Neighbors

Giacomo Bruno[1], Maria Corte-Real Santos[2*], Craig Costello[3], Jonathan Komada Eriksen[4], Michael Meyer[5**], Michael Naehrig[3], and Bruno Sterner[6***]

[1] IKARUS Security Software
giako13@gmail.com
[2] University College London
maria.santos.20@ucl.ac.uk
[3] Microsoft Research, USA
{craigco,mnaehrig}@microsoft.com
[4] Norwegian University of Science and Technology
jonathan.k.eriksen@ntnu.no
[5] University of Regensburg, Germany
michael@random-oracles.org
[6] Surrey Centre for Cyber Security, University of Surrey, UK
b.sterner@surrey.ac.uk

Abstract. We revisit the problem of finding two consecutive B-smooth integers by giving an optimised implementation of the Conrey-Holmstrom-McLaughlin "smooth neighbors" algorithm. While this algorithm is not guaranteed to return the complete set of B-smooth neighbors, in practice it returns a very close approximation to the complete set but does so in a tiny fraction of the time of its exhaustive counterparts. We exploit this algorithm to find record-sized solutions to the pure twin smooth problem, and subsequently to produce instances of cryptographic parameters whose corresponding isogeny degrees are significantly smoother than prior works. Our methods seem well-suited to finding parameters for the SQISign signature scheme, especially for instantiations looking to minimise the cost of signature generation. We give a number of examples, among which are the first parameter sets geared towards efficient SQISign instantiations at NIST's security levels III and V.

Keywords: Post-quantum cryptography, isogeny-based cryptography, twin smooth integers, smooth neighbors, Pell equation, SQISign.

[*] Supported by EPSRC grant EP/S022503/1.

[**] Supported by the German Federal Ministry of Education and Research (BMBF) under the project QuantumRISC (ID 16KIS1039).

[***] Supported by EPSRC grant EP/R513350/1.

1 Introduction

In recent years the tantalising problem of finding two large, consecutive, smooth integers has emerged in the context of instantiating efficient isogeny-based public key cryptosystems. Though the problem was initially motivated in the context of key exchange [9], a wave of polynomial time attacks [6,22,23] has completely broken the isogeny-based key exchange scheme SIDH [19], leaving post-quantum signatures as the most compelling cryptographic application of isogenies at present. In terms of practical potential, the leading isogeny-based signature scheme is SQISign [16]; it boasts the smallest public keys and signatures of all post-quantum signature schemes (by far!), at the price of a signing algorithm that is orders of magnitude slower than its post-quantum counterparts. Finding secure parameters for SQISign is related to the twin smooth problem mentioned above[7], with a large contributing factor to the overall efficiency of the protocol being the smoothness bound, B, of the rational torsion used in isogeny computations. This bound corresponds to the degree of the largest prime-degree isogeny computed in the protocol, for which the fastest algorithm runs in $\tilde{O}(\sqrt{B})$ field operations [4]. Part of the reason for SQISign's performance drawback is that the problem of finding parameters with small B is difficult: the fastest implementation to date targets security comparable to NIST Level I [27, §4.A] and has $B = 3923$ [17]. Additionally, methods for finding efficient SQISign parameters have to date not been able to obtain suitable primes reaching NIST Level III and V security. In view of NIST's recent call for additional general purpose post-quantum signature schemes that are not based on structured lattices [28], it is important to find methods of generating efficient isogeny-based signature parameters beyond those that have been proposed thus far at NIST Level I.

The CHM algorithm. In this work we introduce new ways of finding large twin smooth instances based on the Conrey-Holmstrom-McLaughlin (CHM) "Smooth neighbors" algorithm [8]. For a fixed smoothness bound B, the CHM algorithm starts with the set of integers $S = \{1, 2, \ldots, B-1\}$ representing the smooth neighbors $(1,2), (2,3), \ldots, (B-1, B)$, and recursively grows this set by constructing new twin smooth integers from unordered pairs in $S \times S$ until a full pass over all such pairs finds no new twins, at which point the algorithm terminates. Although the CHM algorithm is not guaranteed to find the set of all B-smooth twins, for moderate values of B it converges with the set S containing *almost all* such twins. The crucial advantage is that, unlike the algorithm of Lehmer [20] that exhaustively solves $2^{\pi(B)}$ Pell equations to guarantee the full set of B-smooth twins, the CHM algorithm terminates much more rapidly. For example, in 2011 Luca and Najman [21] used Lehmer's approach with $B = 100$ to compute the full set of 13,374 twin smooths in 15 days (on a quad-core 2.66 GHz processor) by solving $2^{\pi(B)} = 2^{25}$ Pell

[7] SQISign is instantiated over large primes p such that $p^2 - 1$ is divisible by a large, B-smooth factor. If, for example, we find B-smooth twins r and $r+1$ whose sum is a prime $p = 2r+1$, then $p^2 - 1$ is immediately B-smooth.

equations, the solutions of which can have as many as 10^{10^6} decimal digits. The largest pair of 100-smooth twins they found were the 58-bit integers

$$\begin{aligned} 166055401586083680 &= 2^5 \cdot 3^3 \cdot 5 \cdot 11^3 \cdot 23 \cdot 43 \cdot 59 \cdot 67 \cdot 83 \cdot 89, \text{ and} \\ 166055401586083681 &= 7^2 \cdot 17^{10} \cdot 41^2. \end{aligned}$$

In 2012, Conrey, Holmstrom and McLaughlin ran *their* algorithm on a similar machine to find 13,333 (i.e. all but 41) of these twins in 20 minutes [8]. Subsequently, they set $B = 200$ and found a list of 346,192 twin smooths in about 2 weeks, the largest of which were the 79-bit integers

$$\begin{aligned} 589864439608716991201560 &= 2^3 \cdot 3^3 \cdot 5 \cdot 7^2 \cdot 11^2 \cdot 17 \cdot 31 \cdot 59^2 \cdot 83 \cdot 139^2 \\ &\qquad \cdot 173 \cdot 181, \text{ and} \\ 589864439608716991201561 &= 13^2 \cdot 113^2 \cdot 127^2 \cdot 137^2 \cdot 151^2 \cdot 199^2. \end{aligned}$$

Exhausting the full set of 200-smooth twins would have required solving $2^{\pi(200)} = 2^{46}$ Pell equations, which is pushing the limit of what is currently computationally feasible. The largest run of Lehmer's algorithm reported in the literature used $B = 113$ [9, §5.3], which required solving 2^{30} Pell equations and a significant parallelised computation that ran over several weeks. The largest set of 113-smooth twins found during that computation were the 75-bit integers

$$\begin{aligned} 19316158377073923834000 &= 2^4 \cdot 3^6 \cdot 5^3 \cdot 7 \cdot 23^2 \cdot 29 \cdot 47 \cdot 59 \cdot 61 \cdot 73 \cdot 97 \cdot 103, \\ 19316158377073923834001 &= 13^2 \cdot 31^2 \cdot 37^2 \cdot 43^4 \cdot 71^4. \end{aligned}$$

Remark 1. The above examples illustrate some important phenomena that are worth pointing out before we move forward. Observe that, in the first and third examples, the largest prime not exceeding B is not found in the factors of the largest twins. The largest 89-smooth twins are the same as the largest 97-smooth twins, and the largest 103-smooth twins are the same as the largest 113-smooth twins. In other words, increasing B to include more primes necessarily increases the size of the set of B-smooth twins, but it does not mean we will find any new, larger twins. This trend highlights part of the difficulty we face in trying to find optimally smooth parameters of cryptographic size: increasing the smoothness bound B makes the size of the set of twins grow rapidly, but the growth of the largest twins we find is typically painstakingly slow. The set of 100-smooth twins has cardinality 13,374, with the largest pair being 58 bits; increasing B to 200 gives a set of cardinality (at least) 345,192, but the largest pair has only grown to be 79 bits. In fact, most of this jump in the bitlength of the largest twins occurs when increasing $B = 97$ (58 bits) to include two more primes with $B = 103$ (76 bits). Including the 19 additional primes up to 199 only increases the bitlength of largest twins with $B = 199$ by 3 (79 bits), and this is indicative of what we observe when B is increased even further.

Our contributions. We give an optimised implementation of CHM that allows us to run the algorithm for much larger values of B in order to find larger sized twins. For example, the original CHM paper reported that the full algorithm with $B = 200$ terminated in approximately 2 weeks; our implementation did the same computation in around 943 seconds on a laptop. Increasing the smoothness bound to $B = 547$, our implementation converged with a set of 82,026,426 pairs of B-smooth twins, the largest of which are the 122-bit pair $(r, r+1)$ with

$$\begin{aligned} r &= 5^4 \cdot 7 \cdot 13^2 \cdot 17^2 \cdot 19 \cdot 29 \cdot 41 \cdot 109 \cdot 163 \cdot 173 \cdot 239 \cdot 241^2 \cdot 271 \cdot 283 \\ &\quad \cdot 499 \cdot 509, \qquad \text{and} \\ r+1 &= 2^8 \cdot 3^2 \cdot 31^2 \cdot 43^2 \cdot 47^2 \cdot 83^2 \cdot 103^2 \cdot 311^2 \cdot 479^2 \cdot 523^2. \end{aligned} \tag{1}$$

Although it remains infeasible to increase B to the point where the twins found through CHM are large enough to be used out-of-the-box in isogeny-based schemes (i.e. close to 2^{256}), we are able to combine the larger twins found through CHM with techniques from the literature in order to find much smoother sets of SQISign parameters. In this case we are aided by the requirements for SQISign, which permit us to relax the size of the smooth factor that divides $p^2 - 1$. The current state-of-the-art instantiation [17] uses primes p such that

$$\ell^f \cdot T \mid (p^2 - 1),$$

where ℓ is a small prime (typically $\ell = 2$), where f is as large as possible, and where $T \approx p^{5/4}$ is both coprime to ℓ and B-smooth. For example, the original SQISign implementation [16] used a 256-bit prime p such that

$$p^2 - 1 = 2^{34} \cdot T_{1879} \cdot R,$$

where T_{1879} is an odd 334-bit integer[8] whose largest prime factor is $B = 1879$, and R is the *rough* factor; a 144-bit integer containing no prime factors less than or equal to B. As another example, De Feo, Leroux and Wesolowski [17, §5] instead use a 254-bit prime p with

$$p^2 - 1 = 2^{66} \cdot T_{3923} \cdot R,$$

where T_{3923} is an odd 334-bit integer whose largest prime factor is $B = 3923$, and where all of R's prime factors again exceed B.

During the search mentioned above that found the record 547-smooth twins in (1), over 82 million other pairs of smaller sized twins were found. One such pair was the 63-bit twins $(r-1, r)$ with $r = 8077251317941145600$. Taking $p = 2r^4 - 1$ gives a 253-bit prime p such that

$$p^2 - 1 = 2^{49} \cdot T_{479} \cdot R,$$

[8] The initial SQISign requirements [16] had $T \approx p^{3/2}$, but T_{1879} corresponds to the new requirements.

where T_{479} is an odd 328-bit integer that is 479-smooth. This represents a significant improvement in smoothness over the T values obtained in [16] and [17]. Although the smoothness of T is not the only factor governing the efficiency of the scheme, our analysis in Section 6 suggests that the parameters found in this paper are interesting alternatives to those currently found in SQISign implementations, giving instantiations with a significantly lower expected signing cost, but with a modest increase in verification cost.

Just as we transformed a pair of 85-bit twins into a 255-bit prime by taking $p = 2r^3 - 1$, we combine the use of twins found with CHM and primes of the form $p = 2r^n - 1$ with $n \geq 3$ to obtain several SQISign-friendly primes that target higher security levels. For example, with some 64-bit twins $(r, r+1)$ found through CHM, we give a 382-bit prime $p = 2r^6 - 1$ such that $p^2 - 1 = 2^{80} \cdot T_{10243} \cdot R$, where T is an odd 495-bit integer that is 10243-smooth; this prime would be suitable for SQISign signatures geared towards NIST Level III security. As another example, with some 85-bit twins $(r, r+1)$, we give a 508-bit prime $p = 2r^6 - 1$ such that $p^2 - 1 = 2^{86} \cdot T_{150151} \cdot R$, where T is a 639-bit integer that is 150151-smooth; this prime would be suitable for SQISign signatures targeting NIST Level V security.

Our implementation of the CHM algorithm is written in C/C++ and is found at

`https://github.com/GiacomoBruno/TwinsmoothSearcher`.

Remark 2. In a recent paper [15], it was shown that computing the constructive Deuring correspondence, which is the heavy computation that SQISign needs to perform as part of its signature generation algorithm, is feasible to compute without choosing a specific characteristic p beforehand. However, the paper further confirms (comparing [15, Figure 3] with [15, Table 2]) that the efficiency of this computation depends heavily on the factorisation of $p^2 - 1$ (or more generally $p^k - 1$ for small k). In a setting that allows to freely choose a fixed characteristic p, for instance in the SQISign setting, it is clear that one should choose p carefully for optimal performance.

Remark 3. Another recent work introduces SQISignHD [11], a variant of SQISign in higher dimensions. Although the signature generation could be significantly faster in SQISignHD, the verification algorithm requires computing 4-dimensional isogenies. Since the research of implementing practical 4-dimensional isogenies has mainly only begun since the SIDH attacks, there is no implementation of SQISignHD yet. While breakthroughs in this area of research could change the picture of the field, it remains unclear whether the verification algorithm can be implemented efficiently enough to consider SQISignHD for practical applications, or to reach similar performance as SQISign verification.

Organisation. Section 2 reviews prior methods for generating large instances of twin smooths. In Section 3, we recall the CHM algorithm and give a generalisation of it that may be of independent interest. Section 4 details our implementation of the CHM algorithm and presents a number of optimisations that allowed us to run it for much larger values of B. In Section 5, we discuss the combination of CHM with primes of the form $p = 2x^n - 1$ to give estimates on the probabilities of finding SQISign parameters at various security levels. Section 6 presents our results, giving record-sized twin smooth instances and dozens of SQISign-friendly primes that target NIST's security levels I, III, and V.

2 Preliminaries and Prior Methods

We start by fixing some definitions and terminology.

Definition 1. *A positive integer n is called B-*smooth *for some real number $B > 0$ if all prime divisors of n are at most B. An integer n* generates a B-smooth value *of a polynomial $f(X)$ if $f(n)$ is B-smooth. In this case we call n a B-smooth value of $f(X)$. We call two consecutive integers B-*smooth twins *if their product is B-smooth. An integer n is called B-*rough *if all of its prime factors exceed B.*

We now review prior methods of searching for twin smooth integers by following the descriptions of the three algorithms reviewed in [10, §2] and including the method introduced in [10] itself.

Solving Pell equations. Fix B, let $\{2, 3, \ldots, q\}$ be the set of primes up to B with cardinality $\pi(B)$, and consider the B-smooth twins $(r, r+1)$. Let $x = 2r + 1$, so that $x - 1$ and $x + 1$ are also B-smooth, and let D be the squarefree part of their product $(x-1)(x+1)$, i.e. $x^2 - 1 = Dy^2$ for some $y \in \mathbb{Z}$. It follows that Dy^2 is B-smooth, which means that

$$D = 2^{\alpha_2} \cdot 3^{\alpha_3} \cdot \cdots \cdot q^{\alpha_q}$$

with $\alpha_i \in \{0, 1\}$ for $i = 2, 3, \ldots, q$. For each of the $2^{\pi(B)}$ squarefree possibilities for D, Størmer [24] reverses the above argument and proposes to solve the $2^{\pi(B)}$ Pell equations

$$x^2 - Dy^2 = 1,$$

finding *all* of the solutions for which y is B-smooth, and in doing so finding the complete set of B-smooth twins.

The largest pair of 2-smooth integers is $(1, 2)$, the largest pair of 3-smooth integers is $(8, 9)$, and the largest pair of 5-smooth integers is $(80, 81)$. Unfortunately, solving $2^{\pi(B)}$ Pell equations becomes infeasible before the size of the twins we find is large enough (i.e. exceeds 2^{200}) for our purposes. As we saw in Section 1, [9] reports that with $B = 113$ the largest twins $(r, r+1)$ found upon solving all 2^{30} Pell equations have $r = 19316158377073923834000 \approx 2^{75}$.

The extended Euclidean algorithm. The most naïve way of searching for twin smooth integers is to compute B-smooth numbers r until either $r-1$ or $r+1$ also turns out to be B-smooth. A much better method [9,16] is to instead choose two coprime B-smooth numbers α and β that are both of size roughly the square root of the target size of r and $r+1$. On input of α and β, Euclid's extended GCD algorithm outputs two integers (s,t) such that $\alpha s + \beta t = 1$ with $|s| < |\beta/2|$ and $|t| < |\alpha/2|$. We can then take $\{m, m+1\} = \{|\alpha s|, |\beta t|\}$, and the probability of m and $m+1$ being B-smooth is now the probability that $s \cdot t$ is B-smooth. The reason this performs much better than the naïve method above is that $s \cdot t$ with $s \approx t$ is much more likely to be B-smooth than a random integer of similar size.

Searching with $r = x^n - 1$. A number of works [9,16,17] have found performant parameters by searching for twins of the form $(r, r+1) = (x^n - 1, x^n)$, for relatively small $n \in \mathbb{Z}$. For example, suppose we are searching for b-bit twins $(r, r+1)$ and we take $n = 4$ so that $r = (x^2+1)(x-1)(x+1)$. Instead of searching for two b-bit numbers that are smooth, we are now searching for three smooth $(b/4)$-bit numbers (i.e. $x-1$, x, and $x+1$) and one smooth $(b/2)$-bit number, which increases the probability of success (see [10]).

Searching with PTE solutions. The approach taken in [10] can be viewed as an extension of the method above, where the important difference is that for $n > 2$ the polynomial $x^n - 1$ does not split in $\mathbb{Z}[x]$, and the presence of higher degree terms (like the irreducible quadratic $x^2 + 1$ above) significantly hampers the probability that values of $x^n - 1 \in \mathbb{Z}$ are smooth. Instead, the algorithm in [10] takes $(r, r+1) = (f(x), g(x))$, where $f(x)$ and $g(x)$ are both of degree n and are comprised entirely of linear factors. This boosts the success probability again, but one of the difficulties facing this method is that polynomials $f(x)$ and $g(x)$ that differ by a constant and are completely split are difficult to construct for $n \geq 4$. Fortunately, instances of these polynomials existed in the literature prior to [10], since they can be trivially constructed using solutions to the Prouhet-Tarry-Escott (PTE) problem (see [10]).

3 The CHM Algorithm

In this section, we first recall the Conrey, Holmstrom, and McLaughlin (CHM) algorithm [8], a remarkably simple algorithm that generates twin smooth integers (or *smooth neighbors* as they are called in [8]), i.e. smooth values of the polynomial $X(X+1)$. We then present a generalisation of this algorithm, which generates smooth values of any monic quadratic polynomial. The algorithm generalises the CHM algorithm, as well as another algorithm in the literature by Conrey and Holmstrom [7], which generates smooth values of the polynomial $X^2 + 1$. In the end, we are primarily interested in the CHM algorithm, but present the generalised algorithm here, as it may be of independent interest.

3.1 Finding Smooth Twins with the CHM Algorithm

Conrey, Holmstrom, and McLaughlin [8] present the following algorithm for producing many B-smooth values of $X(X+1)$. It starts with the initial set

$$S^{(0)} = \{1, 2, \ldots, B-1\}$$

of all integers less than B, representing the B-smooth twins $(1,2)$, $(2,3)$, $\ldots$, $(B-1, B)$. Next, it iteratively passes through all pairs of distinct $r, s \in S^{(0)}, r < s$ and computes

$$\frac{t}{t'} = \frac{r}{r+1} \cdot \frac{s+1}{s},$$

writing $\frac{t}{t'}$ in lowest terms. If $t' = t+1$, then clearly t also represents a twin smooth pair. The next set $S^{(1)}$ is formed as the union of $S^{(0)}$ and the set of all solutions t such that $t' = t+1$. Now the algorithm iterates through all pairs of distinct $r, s \in S^{(1)}$ to form $S^{(2)}$ and so on. We call the process of obtaining $S^{(d)}$ from $S^{(d-1)}$ the d-th CHM iteration. Once $S^{(d)} = S^{(d-1)}$, the algorithm terminates.

Example: We illustrate the algorithm for $B = 5$, i.e. with the goal to generate 5-smooth twin integers. The starting set is

$$S^{(0)} = \{1, 2, 3, 4\}.$$

Going through all pairs $(r, s) \in S^{(0)}$ with $r < s$, we see that the only ones that yield a new twin smooth pair $(t, t+1)$ via Equation (2) with t not already in $S^{(0)}$ are $(2,3)$, $(2,4)$ and $(3,4)$, namely,

$$\frac{2}{2+1} \cdot \frac{3+1}{3} = \frac{8}{9}, \quad \frac{2}{2+1} \cdot \frac{4+1}{4} = \frac{5}{6}, \quad \text{and} \quad \frac{3}{3+1} \cdot \frac{4+1}{4} = \frac{15}{16}.$$

Hence, we add 5, 8 and 15 to get the next set as

$$S^{(1)} = \{1, 2, 3, 4, 5, 8, 15\}.$$

The second and third CHM iterations give

$$S^{(2)} = \{1, 2, 3, 4, 5, 8, 9, 15, 24\} \text{ and } S^{(3)} = \{1, 2, 3, 4, 5, 8, 9, 15, 24, 80\}.$$

The fourth iteration does not produce any new numbers, i.e. we have $S^{(4)} = S^{(3)}$, the algorithm terminates here and returns $S^{(3)}$. This is indeed the full set of twin 5-smooth integers as shown in [24], see also [20, Table 1A].

Remark 4. The CHM check that determines whether a pair (r, s) yields an integer solution t to the equation

$$\frac{t}{t+1} = \frac{r}{r+1} \cdot \frac{s+1}{s} \tag{2}$$

can be rephrased by solving this equation for t, which yields

$$t = \frac{r(s+1)}{s-r}. \tag{3}$$

This shows that in order for (r, s) to yield a new pair, $s - r$ must divide $r(s + 1)$ and in particular, must be B-smooth as well.

3.2 Generalising the CHM Algorithm

We now present a generalisation of the CHM algorithm, which finds smooth values of any monic quadratic polynomial $f(X) = X^2 + aX + b \in \mathbb{Z}[X] \subseteq \mathbb{Q}[X]$. The algorithm works with elements in the $\mathbb{Q}$-algebra $A = \mathbb{Q}[X]/\langle f(X)\rangle$. Let $\bar{X}$ denote the residue class of X in A. The generalisation closely follows the idea of the CHM algorithm and is based on the observation that for any $r \in \mathbb{Q}$, we have that

$$N_{A/\mathbb{Q}}(r - \bar{X}) = f(r),$$

where $N_{A/\mathbb{Q}}(\alpha)$ denotes the algebraic norm of $\alpha \in A$ over $\mathbb{Q}$. The algorithm now starts with an initial set

$$S^{(0)} = \{r_1 - \bar{X}, \ldots, r_d - \bar{X}\},$$

where r_i are smooth integer values of $f(X)$ (Definition 1), which means that the element $r_i - \bar{X}$ has smooth non-zero norm. Next, in the d-th iteration of the algorithm, given any two $\alpha, \beta \in S^{(d-1)}$, compute

$$\alpha \cdot \beta^{-1} \cdot N_{A/\mathbb{Q}}(\beta) = r - s\bar{X}$$

for integers r, s (notice that β is invertible, since it has non-zero norm). Now, if s divides r, we obtain an integer $t = \frac{r}{s}$. It follows that

$$\begin{aligned} f(t) &= N_{A/\mathbb{Q}}\left(\frac{r}{s} - \bar{X}\right) \\ &= N_{A/\mathbb{Q}}(r - s\bar{X})s^{-2} \\ &= N_{A/\mathbb{Q}}(\alpha \cdot \beta^{-1} \cdot N_{A/\mathbb{Q}}(\beta))s^{-2} \\ &= N_{A/\mathbb{Q}}(\alpha)N_{A/\mathbb{Q}}(\beta)s^{-2}. \end{aligned}$$

Since both $N_{A/\mathbb{Q}}(\alpha)$ and $N_{A/\mathbb{Q}}(\beta)$ are B-smooth and s is an integer, it follows that t is a B-smooth value of $f(X)$. The set $S^{(d)}$ is then formed as the union of $S^{(d-1)}$ and the set of all such integral solutions. Finally, we terminate when $S^{(d)} = S^{(d-1)}$.

3.3 Equivalence with Previous Algorithms

We now show that the CHM algorithm, as well as another algorithm by Conrey and Holmstrom [7], are special cases of the generalised algorithm, for the polynomials $f(x) = X^2 + X$, and $f(X) = X^2 + 1$ respectively.

Smooth values of $\boldsymbol{X^2 + X}$. To see that the CHM algorithm (see §3.1) is indeed a special case of the generalised algorithm above, we show how the generalised algorithm works for $f(X) = X(X+1) = X^2 + X$. Consider the algebra $A = \mathbb{Q}[X]/\langle X^2 + X\rangle$. This embeds into the matrix algebra $M_{2\times 2}(\mathbb{Q})$ via

$$\psi : r + s\bar{X} \to \begin{pmatrix} r & 0 \\ s & r - s \end{pmatrix}.$$

Instead of working with elements in A, we will work with elements in $\psi(A) \subseteq M_{2\times 2}(\mathbb{Q})$ since this simplifies the argument. In this case, for $\alpha \in A$, we have

$$N_{A/\mathbb{Q}}(\alpha) = \det(\psi(\alpha)).$$

The set corresponding to the initial set in the CHM algorithm is

$$S^{(0)} = \{\left(\begin{smallmatrix} 1 & 0 \\ -1 & 2 \end{smallmatrix}\right), \left(\begin{smallmatrix} 2 & 0 \\ -1 & 3 \end{smallmatrix}\right), \dots, \left(\begin{smallmatrix} B-1 & 0 \\ -1 & B \end{smallmatrix}\right)\}.$$

All these elements clearly have B-smooth norm. The d-th CHM iteration proceeds as follows: For all $\left(\begin{smallmatrix} r & 0 \\ -1 & r+1 \end{smallmatrix}\right), \left(\begin{smallmatrix} s & 0 \\ -1 & s+1 \end{smallmatrix}\right)$ in $S^{(d-1)}$, we try

$$\begin{aligned} \begin{pmatrix} r & 0 \\ -1 & r+1 \end{pmatrix} \begin{pmatrix} s & 0 \\ -1 & s+1 \end{pmatrix}^{-1} s(s+1) &= \begin{pmatrix} r & 0 \\ -1 & r+1 \end{pmatrix} \left(\begin{pmatrix} s+1 & 0 \\ 1 & s \end{pmatrix} \frac{1}{s(s+1)} \right) s(s+1) \\ &= \begin{pmatrix} r(s+1) & 0 \\ -(s-r) & (r+1)s \end{pmatrix}. \end{aligned}$$

Finally, we transform this matrix into the right form, i.e. into a matrix corresponding to an element of the form $\tau = t - \bar{X}$, which means that $\psi(\tau)$ has a -1 in the lower left corner. So, we divide by $s - r$ and end up with the matrix

$$\begin{pmatrix} \frac{r(s+1)}{s-r} & 0 \\ -1 & \frac{(r+1)s}{s-r} \end{pmatrix} = \begin{pmatrix} \frac{r(s+1)}{s-r} & 0 \\ -1 & \frac{r(s+1)}{s-r} + 1 \end{pmatrix}.$$

Now if $\frac{r(s+1)}{s-r}$ is an integer, we add this matrix to the next set $S^{(d+1)}$.

As we have seen in Remark 4, this integer indeed corresponds to the solution (3) of Equation (2) and therefore, the generalised algorithm in the case $f(X) = X^2 + X$ is equivalent to the original CHM algorithm.

Smooth values of X^2+1. Conrey and Holmstrom later presented a method to generate smooth values of X^2+1 [7]. Similar to the CHM algorithm, it starts with an initial set $S^{(0)}$ of positive smooth values of X^2+1. Again, for $d > 0$ and given $r, s \in S^{(d-1)}, r < s$, they compute

$$\frac{rs-1}{s+r}.$$

The next set $S^{(d)}$ is then again formed as the union of $S^{(d-1)}$ and the set of all such values that are integers.

It is equally straightforward to verify that this algorithm is also a special case of the generalised CHM algorithm described above in §3.2. We could again work with matrices in $M_{2\times 2}(\mathbb{Q})$, but here, we are actually working in the number field $K = \mathbb{Q}[X]/\langle X^2+1\rangle$, which is isomorphic to $\mathbb{Q}(i)$, where $i^2 = -1$. The product of the elements $\alpha = r - i$ and $\beta = s - i$ is given as

$$\alpha\beta = (r-i)(s-i) = (rs-1) - (r+s)i.$$

Conrey and Holmstrom's method then simply tries all such products $\alpha\beta$. However, a possibly better choice could be to use

$$\alpha\beta^{-1}N_{K/\mathbb{Q}}(\beta) = \alpha\bar{\beta} = (r-i)(s+i) = (rs+1) - (s-r)i$$

as described in our generalisation. This is due to the fact that the new denominator, $s-r$, is smaller and hence

$$\frac{rs+1}{s-r}$$

is more likely to be an integer[9] (assuming that the numerator follows a random, uniform distribution). As a result, we can expect the algorithm to converge faster.

Whichever option is chosen, one tries to divide by $r+s$ resp. $s-r$, and if the result is an element in $\mathbb{Z}[i]$, it is added to the next set $S^{(d)}$ of smooth values of X^2+1. Conrey and Holmstrom's method is therefore another special case of the generalised algorithm.

Remark 5. We note that neither the generalised CHM algorithm, nor any of the previous special cases give any guarantees to what proportion of B-smooth values of $f(X)$ it finds. However, for the previous special case algorithms, certain conjectural results have been stated, based on numerical evidence, which suggests that the algorithm returns all but a small fraction of all smooth values of the respective quadratic polynomials. We make no similar claims for the general case algorithm.

[9] Another alternative is to include both positive and negative values in the inital set $S^{(0)}$. Observe that in this case, it does not matter whether one uses $(rs+1)/(s-r)$ or $(rs-1)/(s+r)$, as $(rs+1)/(s-r) = -(s(-r)+1)/(s+(-r)))$.

4 Searching for Large Twin Smooth Instances: CHM in Practice

Ideally, the CHM algorithm could be run as described in [8] with a large enough smoothness bound B to find twin smooths of cryptographic sizes. However, experiments suggest that this is not feasible in practice. We report on data obtained from an implementation of the pure CHM algorithm in §4.1, present several optimisations in §4.2 and details on our optimised implementation in §4.3.

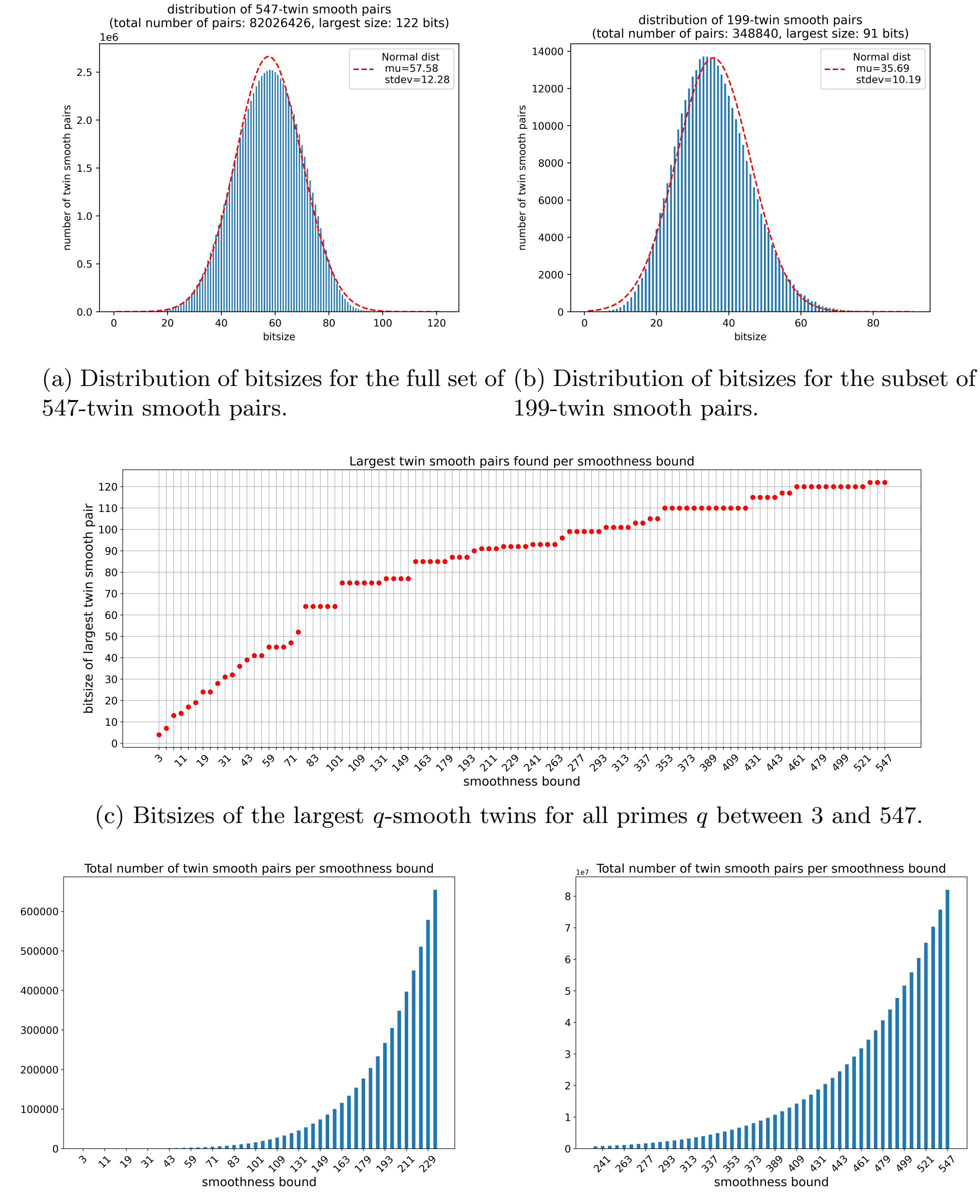
4.1 Running CHM in Practice

In order to collect data and assess the feasibility of finding large enough twin smooths, we implemented a somewhat optimised version of the pure CHM algorithm. In particular, this implementation is parallelised, and avoids multiple checks of the same pairs of twin smooths (r, s). Furthermore, we iterate through smoothness bounds: We start with a small bound B_1 and the initial set $S_1^{(0)} = \{1, \ldots, B_1 - 1\}$, and use the CHM algorithm to iteratively compute sets $S_1^{(i)}$ until we reach some d_1 such that $S_1^{(d_1)} = S_1^{(d_1-1)}$. In the next iteration, we increase the smoothness bound to $B_2 > B_1$ and define the initial set $S_2^{(0)} = S_1^{(d_1)} \cup \{B_1, \ldots, B_2 - 1\}$. Again we compute CHM iterations until we find d_2 such that $S_2^{(d_2)} = S_2^{(d_2-1)}$, where we avoid checking pairs (r, s) that have been processed in earlier iterations. Ideally, we could repeat this procedure until we reach a smoothness bound B_i for which the CHM algorithm produces large enough twin smooths for cryptographic purposes. However, our data suggests that this is infeasible in practice due to both runtime and memory limitations.

In particular, we ran this approach up to the smoothness bound $B = 547$, and extrapolating the results gives us rough estimations of the largest possible pair and number of twin smooths per smoothness bound.

After the $B = 547$ iteration, the set of twin smooths contains 82,026,426 pairs, whose bitlength distribution roughly resembles a normal distribution centered around bitlength 58. The largest pair has a bitlength of 122 bits. An evaluation of the obtained set is shown in Figure 1. Figure 1a shows the distribution of bitsizes in the full set, while Figure 1b shows that of the subset of all 199-smooth twins obtained in this run. Figure 1c shows the bitsize of the largest q-smooth twin pairs for each prime q between 3 and 547. And Figures 1d and 1e show the number of q-smooth twins for each such q.

Using the data of these experiments, we can attempt to estimate at which smoothness bound B this approach can be expected to reach twin smooths of cryptographic sizes, and how much memory is required to run iterations to reach this B. The data visualised in Figure 1cindicates that the bound necessary for the largest twin smooth pair obtained by running CHM with this bound to reach a bitlength of 256 lies in the thousands, possibly larger than 5,000. Similarly, the data displayed in Figures 1d and 1eshows how quickly the number of B-smooth twins increases with B. Given that the effort for CHM iterations



(a) Distribution of bitsizes for the full set of 547-twin smooth pairs.

(b) Distribution of bitsizes for the subset of 199-twin smooth pairs.

(c) Bitsizes of the largest q-smooth twins for all primes q between 3 and 547.

(d) Number of q-smooth twins for all primes q between 3 and 233.

(e) Number of q-smooth twins for all primes q between 239 and 547.

Fig. 1: Evaluation of the set of 547-smooth twins obtained by running the original CHM algorithm with smoothness bound $B = 547$. The bitsize of a pair $(r, r+1)$ is $\lfloor \log r \rfloor + 1$. Data for the number of q-smooth twins for all primes q up to 547 has been split into two histograms of different scale.

grows quadratically with the set size, these estimates indicate that it is not feasible to reach cryptographically sized smooth twins with the original CHM algorithm.

4.2 Optimisations

One major issue with running the plain CHM algorithm for increasing smoothness bound is the sheer size of data that needs to be dealt with. The sets $S_i^{(d_i)}$ grow very rapidly and the quadratic complexity of checking all possible pairs (r, s) leads to a large runtime. The natural question that arises is whether CHM can be restricted to checking only a certain subset of such pairs without losing any or too many of the new smooth neighbors. Furthermore, if the purpose of running the CHM algorithm is not to enumerate all twin smooth pairs for a given smoothness bound but instead, to produce a certain number of pairs of a given size or to obtain some of the largest pairs, it might even be permissible to omit a fraction of pairs.

To find a sensible way to restrict to a smaller set, we next discuss which pairs (r, s), $r < s$ result in a given twin smooth pair $(t, t + 1)$ via

$$\frac{r}{r+1} \cdot \frac{s+1}{s} = \frac{t}{t+1}. \tag{4}$$

This is discussed in [8, §3], but we elaborate on it in a slightly different way here. Let $t > 0$, let u be any divisor of t and v any divisor of $t+1$. Let $h, x \in \mathbb{Z}$ be given by $t = uh$ and $t+1 = vx$ (where $u, v, h, x > 0$). Therefore, $v/u = h/x + 1/(ux)$. If $u < v$ then $h > x$ and if $u > v$ then $h < x$. We therefore fix $u < v$ (otherwise switch the roles of u, v and h, x). Since $u < v$, the pair

$$(r, s) = (t - \frac{u}{v}(t+1),\ \frac{v}{u}t - (t+1) = \frac{v}{u}r) \tag{5}$$

satisfies Equation (4) and it follows that

$$r = u(h - x),\ r + 1 = x(v - u),\ s = v(h - x),\ s + 1 = h(v - u). \tag{6}$$

Therefore, $s/r = v/u$ and $(s + 1)/(r + 1) = h/x$, $u < v$, $h > x$ and $0 < r < s$. This also means that $s = r + (v - u)(h - x)$, $t = r + ux$ and that $\gcd(r(s + 1), s(r + 1)) = s - r = (v - u)(h - x)$ (note that $\gcd(uh, vx) = \gcd(t, t + 1) = 1$).

Conversely, given (r, s) with $r > 0$ that satisfy Equation 4, define $u = r/\gcd(r, s)$ and $v = s/\gcd(r, s)$, then $s > r$, $u < v$ and $u \mid t$, $v \mid (t + 1)$. Hence we have the correspondence between the set of pairs (r, s) with $r < s$ that yield a new twin pair $(t, t + 1)$ via Equation (4) and the set of pairs of divisors of t and $t + 1$ described in [8, §3] as follows:

$$\begin{aligned}&\{(r, s) \mid r < s \text{ and } r(s + 1)(t + 1) = s(r + 1)t\}\\ &\qquad \longleftrightarrow \{(u, v) \mid u < v \text{ and } u \mid t,\ v \mid (t + 1)\}.\end{aligned} \tag{7}$$

However, this correspondence does not identify the pairs (r, s) corresponding to twin smooths, i.e. given (u, v) there is no guarantee that any of $r, r+1, s, s+1$ are B-smooth. This is not discussed in [8, §3]. The next lemma fills this gap by stating an explicit condition on the divisors u, v, h, x.

Lemma 1. *Let $t \in \mathbb{Z}$ such that $t(t+1)$ is B-smooth. Let (u, v) be a pair of divisors such that $t = uh$, $t+1 = vx$ and let (r, s) be defined as in Equation (5).*

Then $r(r+1)s(s+1)$ is B-smooth if and only if $(v-u)(h-x) = s-r$ is B-smooth.

Proof. As divisors of t and $t+1$, u and v as well as h and x are all B-smooth. The statement follows from the Equations (6). □

Using similar sized pairs. We next consider the following condition to restrict the visited pairs (r, s) in CHM as a mechanism to reduce the set size and runtime. Let $k > 1$ be a constant parameter. We then only check pairs (r, s) if they satisfy

$$0 < r < s < kr. \tag{8}$$

Assume that (r, s) results in a pair $(t, t+1)$ through satisfying Equation (4). As seen above, $\frac{s}{r} = \frac{v}{u}$ for $u \mid t$, $v \mid (t+1)$, so we can use (u, v) to determine which values k are useful. Since $\frac{v}{u} < k$, it follows $s = \frac{v}{u}t - (t+1) < (k-1)t$. If we are only interested in obtaining a new t from a pair (r, s) such that $s < t$, we can take $k \leq 2$, overall resulting in $1 < k \leq 2$.

This k seems to be a good quantity to study as we can relate it to the factors of $v - u$. Indeed, $v - u = u(\frac{v}{u} - 1) = u(\frac{s}{r} - 1)$ and we have $s < kr$.

Definition 2. *Let $(r, r+1)$ and $(s, s+1)$ be twin smooths with $r < s$ and $k \in \mathbb{R}$ with $1 < k \leq 2$. We call the pair (r, s)* k-balanced *if $r < s < k \cdot r$.*

We want to find a k such that a k-balanced pair (u, v) subject to the above conditions will yield a balanced r, s such that $r, r+1, s, s+1$ are B-smooth, or equivalently that $v - u$ and $h - x$ are.

Running the CHM algorithm only with 2-balanced pairs (r, s) then guarantees that any t produced by Equation 4 will be larger than the inputs r and s. Although we sacrifice completeness of the set of twin B-smooths with this approach, we can significantly reduce the runtime.

We can even push this approach further. Recall that we require $\gcd(r(s+1), (r+1)s) = s - r$ in order to generate a new pair of twin smooths $(t, t+1)$. By Lemma 1, this can only hold if $\Delta = s - r$ is B-smooth. Hence, only checking pairs (r, s) for which Δ is likely to be smooth increases the probability for a successful CHM step. Heuristically, the smaller Δ is, the better the chances for Δ to be smooth. Furthermore, if Δ contains small and only few prime factors, the probability for the condition $\Delta = \gcd(r(s+1), (r+1)s)$ is relatively high. We can summarise this in the following heuristic.

Heuristic 1 *Let $k_1, k_2 \in \mathbb{R}$ with $1 < k_1 < k_2 \leq 2$, and (r_1, s_1) resp. (r_2, s_2) a k_1- resp. k_2-balanced pair of twin smooths. Then the probability for (r_1, s_1) to generate new twin smooths via the CHM equation is larger than that for (r_2, s_2).*

In order to save additional runtime, we can thus pick k closer to 1, and only check the pairs (r, s) that are most likely to generate new twin smooths. Therefore, we can still expect to find a significant portion of all twin B-smooths for a given smoothness bound B. We expand on the choice of k and different ways of implementing this approach in §4.3.

Thinning out between iterations. Another approach to reduce both runtime and memory requirement is to thin out the set of twin smooths between iterations. In particular, once we finished all CHM steps for a certain smoothness bound B_i, we can remove twins from the set $S_i^{(d_i)}$ based on their likeliness to produce new twin smooths before moving to the next iteration for B_{i+1}.

One possible condition for removing twins is to look at their smoothness bounds. Let $(r, r+1)$ be B_1-smooth, $(s, s+1)$ be B_2-smooth (but not B-smooth for any $B < B_2$), and $B_1 \ll B_2$. Since $(s, s+1)$ contains (multiple) prime factors larger than B_1, they cannot be contained in $(r, r+1)$, which makes the requirement $\gcd(r(s+1), (r+1)s) = s - r$ heuristically less likely to be satisfied. However, in practice it turns out that the differences between the smoothness bounds we are concerned with are not large enough for this heuristic to become effective.

In our experiments, it turned out to be more successful to keep track of how many new twin smooths each r produces. We can then fix some bound m, and discard twins that produced less then m twins after a certain number of iterations. Our experiments suggest that using this approach with carefully chosen parameters yields a noticeable speedup, but fails completely at reducing the memory requirements, as we still need to keep track of the twins we already found. Furthermore, in practice the approach of only using k-balanced twins turned out to be superior, and hence we focus on this optimisation in the following.

4.3 Implementation

We implemented the CHM algorithm with several of the aforementioned optimisations in C++, exploiting the fact that it parallelises perfectly. Note that some of our approaches require the set of twin smooths to be sorted with respect to their size. Hence, an ordered data structure is used for storing the twins set. We used the following techniques and optimisations.

CHM step. For each pair (r, s) considered by the implementation, we have to check if Equation (4) holds. As mentioned in §4.2, this requires that $\gcd(r(s+1), (r+1)s) = s - r$ is satisfied. However, we can completely avoid the gcd calculation by observing that we require $r \cdot (s+1) \equiv 0 \mod (s-r)$. Only if this is the case we perform a division to compute t, which represents the new pair of twin smooths $(t, t+1)$. Therefore, we only perform one modular reduction per considered pair (r, s), followed by one division if the CHM step is successful. This is significantly cheaper than a naïve implementation of Equation (4) or a gcd computation.

Data structure. Initially the set of twins was organised in a standard C array, that each time an iteration completed was reallocated to increase its size, and reordered.

To avoid the overall inefficiency of this method we moved to use the C++ standard library std::set. This data structure is implemented with a Red Black tree, guarantees $O(\log N)$ insertion and search, while keeping the elements always ordered.

We then moved to use B+Trees [5], that have the same guarantees for insertion, search, and ordering, but are more efficient in the memory usage. Because the elements of a B+Tree are stored close to each other in memory it becomes much faster to iterate through the set, an operation that is necessary for creating the pairs used in each computation.

Implemented optimisations. As discussed in §4.2, we focus on the case of k-balanced pairs (r, s), which satisfy $r < s < k \cdot r$. Compared to the full CHM algorithm, this leads to a smaller set of twin smooths, but allows for much faster running times. We implemented the k-balanced approach in various different flavours.

Global-k. In the simplest version - the `global-`k approach - we initially pick some k with $1 < k \leq 2$, and restrict the CHM algorithm to only check k-balanced pairs (r, s). The choice of k is a subtle manner: Picking k too close to 1 may lead to too many missed twin smooths, such that we cannot produce any meaningful results. On the other hand, picking k close to 2 may result in a relatively small speedup, which does not allow for running CHM for large enough smoothness bounds B. Unfortunately, there seems to be no theoretical handle on the optimal choice of k, which means that it has to be determined experimentally. We note that when picking an aggressive bound factor $k \approx 1$, small numbers r in the set of twins S may not have any suitable $s \in S$ they can be checked with. Thus, we pick a different bound, e.g. $k = 2$, for numbers below a certain bound, e.g. for $r \leq 2^{20}$.

Iterative-k. Instead of iterating through smoothness bounds B_i as described in §4.1 and using the `global-`k approach, we can switch the roles of B and k if we are interested in running CHM for a fixed smoothness bound B. We define some initial value k_0, a target value $k_{\max}$, and a step size $k_{\text{step}} > 0$. In the first iteration, we run CHM as in the `global-`k approach, using k_0. The next iteration then increases to $k_1 = k_0 + k_{\text{step}}$, and we add the condition to not check pairs (r, s) if they were already checked in previous iterations. We repeat this iteration step several times until we reach $k_{\max}$. Compared to the `global-`k approach, this allows us to generate larger B-smooth twins faster, since we restrict to the pairs (r, s) first that are most likely to generate new twins. However, the additional checks if previous pairs have been processed in earlier iterations add a significant runtime overhead. Thus, this method is more suitable for finding well-suited choices of k, while actual CHM searches benefit from switching to the `global-`k approach.

Constant-range. In both the `global-`k and `iterative-`k approach, the checks if a pair (r, s) is k-balanced, or has been processed in earlier iterations, consumes a significant part of the overall runtime. Therefore, we can use constant ranges to completely avoid these checks. Since we always keep the set of twins S sorted by size, the numbers s closest to r (with $s > r$) are its neighbors in S. Thus, we can sacrifice the exactness of the k-balanced approaches above, and instead fix a range R and for each r check (r, s) with the R successors s of r in S. As shown below, this method significantly outperforms the `global-`k approach due to the elimination of all checks for k-balance. This is true even when R is large enough to check more pairs than are considered in the `global-`k approach for a given k.

Variable-range. Similar to the `constant-range` approach, we can adapt the range R depending on the size of r. For instance, choosing r at the peak of the size distribution will lead to many possible choices of s such that (r,s) are balanced. Hence, we can choose a larger range R whenever more potential pairs exist, while decreasing R otherwise. In practice, the performance of this method ranks between `global-`k and `constant-range` by creating roughly the same pairs that `global-`k creates without any of the overhead of the balance checks. If R is chosen large enough such that the `constant-range` approach ends up generating more pairs than `global-`k, then `variable-range` performs better. Realistically, the size of the range R increases by (very) roughly 3% for each prime number smaller than the smoothness bound B, and slows down the algorithm drastically at higher smoothness, similarly to the k-based approaches.

Remark 6. Similar to the `variable-range` approach, we experimented with a variant of the `global-`k approach, which adjusts k according to the size of r to find suitable s for the CHM step. However, the `constant-range` and `variable-range` approaches turned out to be superior in terms of performance, and therefore we discarded this `variable-`k variant.

Performance comparison. In order to compare the implications of the optimisations in practice, we ran different variants of the CHM implementation for the fixed smoothness bound $B = 300$. All experiments ran on a machine configured with 4 x Xeon E7-4870v2 15C 2.3 GHz, 3072 GB of RAM. The total amount of parallel threads available was 120. As described above, the `global-`k and `constant-range` approach significantly outperform their respective variants, hence we focus on different configurations of these two methods.

The results are summarised in Table 1. For both the `global-`k and the `constant-range` approach we measured the results for conservative and more aggressive instantiations,

Variant	Parameter	Runtime	Speedup	#twins	#twins from largest 100
Full CHM	-	4705s	1	2300724	100
`global-`k	$k = 2.0$	364s	13	2289000	86
	$k = 1.5$	226s	21	2282741	82
	$k = 1.05$	27s	174	2206656	65
`constant-range`	$R = 10000$	82s	57	2273197	93
	$R = 5000$	35s	134	2247121	87
	$R = 1000$	16s	294	2074530	75

Table 1: Performance results for different variants of our CHM implementation for smoothness bound $B = 300$. Speedup factors refer to the full CHM variant.

where smaller values of k and R are considered more aggressive. It is evident that already for the conservative instantiations, we gain significant performance speedup, while still finding almost the full set of twin smooths, and most of the 100 largest 300-smooth twins. For the more aggressive instantiations, we miss more twins, yet still find a significant amount of large twins.

As discussed above, the `constant-range` approach outperforms the `global-`k approach in terms of runtime, due to the elimination of all checks for k-balance of twins. Interestingly, while very aggressive instantiations of `constant-range` miss more twin smooths, they find a larger share of the largest 100 twins than their `global-`k counterpart. Therefore, we conclude that for larger smoothness bounds B, for which we cannot hope to complete the full CHM algorithm, `constant-range` is the most promising approach for obtaining larger twin smooths within feasible runtimes.

Remark 7. While all optimisations lose a small proportion of the largest twin smooths, they are not necessarily lost permanently. In practice, when iterating to larger smoothness bounds B_i, we often also find some B_j-smooth twins for bounds $B_j < B_i$. Thus, the size of the set of 300-smooth twins usually increases in the optimised variants when moving to larger B.

Remark 8. In the following sections, we will require twin smooths of a certain (relatively small) bitlength. This can easily be incorporated into all implemented variants by removing all twins above this bound after each iteration. This means that we cut off the algorithm at this size, and do not attempt to obtain larger twins, which significantly improves the runtime and memory requirements.

n	$p_n(x)^2 - 1$
2	$4x^2(x-1)(x+1)$
3	$4x^3(x-1)(x^2+x+1)$
4	$4x^4(x-1)(x+1)(x^2+1)$
5	$4x^5(x-1)(x^4+x^3+x^2+x+1)$
6	$4x^6(x-1)(x+1)(x^2-x+1)(x^2+x+1)$

Table 2: Factorisation of $p_n(x)^2 - 1$ for $n = 2, 3, 4, 5, 6$, where $p_n(x) = 2x^n - 1$

5 Fantastic p's and where to find them: Cryptographic Primes of the form $p = 2r^n - 1$

This section focuses on finding primes suitable for isogeny-based cryptographic applications. As discussed in the previous sections, the pure CHM method does not allow for us to directly compute twins of at least 256 bits as required for this aim. However, some cryptographic applications, for example the isogeny-based signature scheme SQISign, do not need twins $(r, r+1)$ that are fully smooth. Indeed, the current incarnation of SQISign requires a prime p that satisfies $2^f T \mid p^2 - 1$, where f is as large as possible, and $T \approx p^{5/4}$ is smooth and odd [17]. This flexibility allows us to move away from solely using CHM and, instead, to use CHM results as inputs to known methods for finding such primes. At a high level, we will find fully smooth twins of a smaller bit-size via CHM and boost them up using the polynomials $p_n(x) = 2x^n - 1$ (for carefully chosen n). Hence, if $r, r+1$ are fully smooth integers and n is not too large, we can guarantee a large proportion of $p_n(r)^2 - 1$ to be smooth.

Notation. For a variable x, we will denote $2x^n - 1$ by $p_n(x)$, and the evaluated polynomial $p_n(r)$ by p, emphasising that it is an integer.

General method. In this section, we will give a more in-depth description of the approach to obtaining cryptographic sized primes p, such that $p^2 - 1$ has $\log T'$ bits of B-smoothness, where $T' = 2^f T$. We recall that for our SQISign application, we have $\log p \in \{256, 384, 512\}$ for NIST Level I, III and V (respectively), $T \approx p^{5/4}$ and f as large as possible. In the current implementation of SQISign, $f \approx \lfloor \log\left(p^{1/4}\right) \rfloor$ (i.e., $T' \approx p^{3/2}$), and therefore, we aim for this when finding primes.

Fix a smoothness bound B and let $p_n(x) = 2x^n - 1$. We have $p_n(x)^2 - 1 = 4x^n(x-1)f(x)$ for some polynomial $f(x)$, as shown in Table 2.

We observe that for n even, both $x+1$ and $x-1$ appear in the factorisation of $p_n(x)^2 - 1$. In this case, for twin smooths $(r, r \pm 1)$, evaluating $p_n(x)$ at r guarantees that we have a smooth factor $4x^n(x \pm 1)$ in $p^2 - 1$. For n odd, we will only have that $x - 1$

appears in the factorisation, and therefore only consider twins $(r, r-1)$ to guarantee we have B-smooth factor $4x^n(x-1)$.

The first step is to use our implementation of the CHM algorithm, described in Sections 3 and 4, to obtain B-smooth twins $(r, r \pm 1)$ of bitsize approximately $(\log p - 1)/n$. We then obtain primes of suitable sizes via computing $p = p_n(r)$ for all candidate r, as described above. By construction, $p^2 - 1$ has guaranteed $\frac{n+1}{n}(\log(p) - 1) + 2$ bits of smoothness. We then require that the remaining factors have at least

$$\max\left(0,\ \frac{3}{2}\log p - \left(\frac{n+1}{n}(\log p - 1) + 2\right)\right)$$

bits of B-smoothness. In Section 5.2, we will discuss the probability obtaining this smoothness from the remaining factors.

5.1 Choosing n

For small n, we require CHM to find twin smooths of *large* bit size. For certain bit sizes, running full CHM may be computationally out of reach, and therefore we use a variant that may not find all twins. In this case, however, we have more guaranteed smoothness in $p^2 - 1$ and so it is more likely that the remaining factors will have the required smoothness. For large n, we can obtain more twin smooths from CHM (in some cases, we can even exhaustively search for all twin smooths), however we have less guaranteed smoothness in $p^2 - 1$. Finding values of n that balance these two factors will be the focus of this section.

$n = 2$. Let $(r, r \pm 1)$ be twin smooth integers and let $p = 2r^2 - 1$. In this case, $2r^2(r \pm 1) \mid T'$, meaning that $\log T' \geq \frac{3}{2}\log p$, and we have all the required smoothness. Write $T' = 2^f T = 2r^2(r \pm 1)$ where T is odd. If $f < \lfloor \log\left(p^{1/4}\right) \rfloor$, we have $T > p^{5/4}$, and we do not have to rely on a large power of 2 dividing $r - 1$. Otherwise, we turn to Section 5.2 to estimate the probability of $r \mp 1$ having enough small factors to make up for this difference.

Suppose we target primes with λ bits of classical security, i.e., we need a prime of order $p \approx 2^{2\lambda}$. For $n = 2$, this corresponds to finding twin smooths of size $\approx 2^{\lambda - \frac{1}{2}}$, and so is only suitable for finding NIST Level I parameters due to the limitations of the CHM method (see Section 4). One could instead use other techniques for finding large enough twins for $n = 2$, such as the PTE sieve [10], at the cost of significantly larger smoothness bounds. Alternatively, we can move to higher n, which comes at the cost of loosing guaranteed smoothness. Another challenge here is that, given the relatively large size of the twins, it appears difficult to find enough twins for obtaining primes with a large power of two.

$n = 3$. Let $(r, r-1)$ be twin smooth integers and let $p = 2r^3 - 1$. Here, we can guarantee that the smooth factor T' of $p^2 - 1$ is at least of size $\approx p^{4/3}$. If $f < \lfloor \log_2\left(p^{1/12}\right) \rfloor$, we have $T > p^{5/4}$. Otherwise, we require that there are enough smooth factors in $r^2 + r + 1$ to reach this requirement.

Here, for λ bits of classical security, we need to target twin smooth integers of size $\approx 2^{\frac{2\lambda-1}{3}}$. In this case, the CHM method will (heuristically) allow us to reach both NIST Level I and III parameters.

$n = 4$. Let $(r, r \pm 1)$ be twin smooth integers and $p = 2r^4 - 1$. Here we can only guarantee a factor of size $\approx p^{5/4}$ of $p^2 - 1$ to be smooth. When accounting for the power of two, we must hope for other smooth factors. As $p_n(x) - 1$ splits into (relatively) small degree factors, namely $p_n(x) - 1 = 2(x-1)(x+1)(x^2+1)$, the probability of having enough B-smooth factors is greater (than if there was, for example, a cubic factor).

In contrast to the previous cases, this setting should be suitable for targeting all necessary security parameters. However, for the NIST Level I setting, the work by De Feo, Leroux and Wesolowski [17][§5.2] showed that the best one could hope for here while maximising the power of two gives SQISign parameters with a smoothness bound of ≈ 1800. While this is a better smoothness bound than the NIST Level I prime with the best performance for SQISign, it does not perform as well in practice. Indeed, most of the odd primes less than 1800 that appear in $p^2 - 1$ are relatively large, making isogeny computation relatively slow. In the best performing prime, however, a large power of 3 divides $p^2 - 1$, and most of its other odd prime divisors are fairly small. We note that the authors of [17] only searched for parameters that maximise the power of two, and hence there could be some scope to find parameters that have slightly smaller powers of two.

Other n. For larger n, the amount of guaranteed smoothness decreases, and thus the probability that the remaining factors have the required smoothness is small. Indeed, we find that only $n = 6$ has the correct balance of requiring small twin smooths while still having a reasonable probability of success. This is primarily due to the factorisation of $p_6(x) - 1 = 2(x-1)(x+1)(x^2 - x + 1)(x^2 + x + 1)$, having factors of degree at most 2, which improves the probability that we have enough smooth factors. In contrast, $n = 5$ results in more guaranteed smoothness than $n = 6$, but requires the quartic factor in $p_5(x) - 1$ to provide the necessary smoothness, which is relatively unlikely.

While one could use $n = 6$ to find NIST Level I parameters, this larger n shines in its ability to give us both NIST Level III and V parameters.

5.2 Probability of Sufficient Smoothness

In this section, we determine the probability of obtaining cryptographic primes with sufficient smoothness using the methods outlined above. We follow Banks and Shparlinski [1] to determine the probability of $p^2 - 1$ being sufficiently smooth for some prime p. More precisely, given that the factor $r(r \pm 1) \mid p^2 - 1$ is already fully smooth, we want to calculate the probability of $p^2 - 1$ having $\log T'$-bits of B-smoothness.

First, we find the probability that the factor $r(r \pm 1) \mid p^2 - 1$ is fully smooth, i.e., the probability of finding fully B-smooth twins $(r, r \pm 1)$. To do so, we use the following counting function:

$$\Psi(X, B) = \#\{N \le X : N \text{ is } B\text{-smooth}\}.$$

For a large range of X and B, it is known that

$$\Psi(X, B) \sim \rho(u)X,$$

where $u = (\log X)/(\log B)$ and ρ is the Dickman function [14,12]. The Dickman function is implemented in most computational algebra packages, including SageMath, which allows us to evaluate $\Psi(X, B)$ for various X and B. In practice, we find B-smooth twins $(r, r \pm 1)$ using our implementation of the CHM algorithm as described in 4.

Next, we calculate the probability of $p^2 - 1$ having $\log T'$-bits of B-smoothness. As $p^2 - 1$ may only be partially smooth, we will use the following counting function

$$\Theta(X, B, D) = \#\{N \le X : D < \text{largest } B\text{-smooth divisor of } N\}.$$

The value $\Theta(X, B, D)$ will give the number of positive integers $N \le X$ for which there exists a divisor $d \mid N$ with $d > D$ and such that d is B-smooth. This function has been previously studied in the literature, for example [26,25]. For X, B, D varying over a wide domain, Banks and Shparlinski [1, Theorem 1] derive the first two terms of the asymptotic expansion of $\Theta(X, B, D)$. By implementing this expansion, we are able to estimate the value of Θ at various X, B, D in the correct range.

As discussed in the section above, we restrict to $n = 2, 3, 4, 6$. Recall that $p_n(x)^2 - 1 = 4x^n(x-1)f(x)$, as given in Table 2 for each $2 \le n \le 6$. Write $f(x) = f_1(x) \cdots f_k(x)$, where each f_i is irreducible of degree $d_i = \deg(f_i)$ and $d = \deg(f)$. To calculate the probabilities, we require that the probability of $f(x)$ having at least $\log_2 D$-bits of B-smoothness is the product of the probabilities of each of its factors f_i having at least $\log_2 D_i$-bits of B-smoothness where $\log_2 D = \sum_{i=1}^{k} \log_2 D_i$. We can view this as an extension of [10, Heuristic 1]. Note that the only constraint on how the smoothness is distributed between the factors $f_i(x)$ is that the total bit size of B-smooth factors must equal $\log_2 D$. We could, for example, sum over all the possible distributions of smoothness using the inclusion-exclusion principle. However, in distributions where one of the factors has a very small amount of smoothness, we fall out of the ranges allowed as input into Θ determined by [1, Theorem 1]. Therefore, for simplicity, we will assume that smoothness is distributed evenly between the remaining factors (weighted by the degree), i.e., $\log_2 D_i = (d_i \log_2 D)/d$. In reality, this only gives us a lower bound for the probability, but this will suffice for our purposes. Obtaining a more theoretical and accurate grasp on these probabilities is left as an avenue for future research.

In Table 3, we give an overview of the relevant probabilities for NIST Level I, III, and V parameters, calculated as described above. We observe that as n gets larger, the probability of finding B-smooth integers of the appropriate bitsize increases. In contrast,

	n	$\log_2(r)$	**Probability of B-smooth** $(r, r \pm 1)$	**Probability of $p^2 - 1$** $\log T'$**-bits B-smooth given** $(r, r \pm 1)$ **twin smooth**	**Extra Smoothness Needed**
NIST-I	2	≈ 127.5	$2^{-58.5}$	1	0
$B = 2^9$	3	≈ 85.0	$2^{-32.1}$	$2^{-8.4}$	42
$\log p = 256$	4	≈ 63.8	$2^{-20.5}$	$\approx 2^{-12.7}$	63.3
$\log T' = 384$	6	≈ 42.5	$2^{-10.4}$	$\approx 2^{-16.8}$	84.5
NIST-III	2	≈ 191.5	$2^{-55.7}$	1	0
$B = 2^{14}$	3	≈ 127.7	$2^{-30.5}$	$2^{-8.2}$	63.3
$\log p = 384$	4	≈ 95.8	$2^{-19.4}$	$\approx 2^{-12.4}$	95.3
$\log T' = 576$	6	≈ 63.8	$2^{-9.7}$	$\approx 2^{-16.2}$	127.2
NIST-V	2	≈ 255.5	$2^{-63.7}$	1	0
$B = 2^{17}$	3	≈ 170.3	$2^{-35.2}$	$2^{-9.6}$	84.7
$\log p = 512$	4	≈ 127.8	$2^{-22.6}$	$\approx 2^{-14.5}$	127.3
$\log T' = 768$	6	≈ 85.2	$2^{-11.5}$	$\approx 2^{-19.2}$	169.8

Table 3: Assuming that $(r, r \pm 1)$ are twin smooth integers and p has $\log p$ bits, calculates the probability of having a B-smooth divisor $T' \mid p^2 - 1$ of size $\approx p^{3/2}$. More details in text.

for bigger n we are guaranteed less smoothness in $p^2 - 1$. As a result, given B-smooth twins, the probability of finding a SQISign prime p decreases as n increases. For each NIST level, we predict that the n that balance these two contrasting probabilities have a higher chance of finding a p satisfying our requirements. As discussed in the next section, this trend is reflected in practice.

6 Results and Comparisons

In this section we give the concrete results that were obtained from our experiments with the CHM algorithm, and analyse the various twins in relation to SQISign in accordance with the relevant bitsizes mentioned in Table 3.

6.1 Record Twin Smooth Computations

We ran the optimised full CHM algorithm with $B = 547$ and found a total of 82,026,426 pairs of B-smooth twins. Among these pairs, we found 2,649 additional 200-smooth twins that were not found by the original authors of the algorithm [8]. This showcases the validity of Remark 5 that the algorithm does not guarantee us to find all B-smooth twins. Furthermore, there is no guarantee that running CHM with $B = 547$ will produce all 200-smooth twins. As mentioned in the introduction, the only way to see how far away we are from the exact number of 200-smooth twins is to solve all 2^{46} Pell equations.

For the application mentioned in the previous section, we only need twins of a certain bitsize. Within this set of twins, 9,218,648 pairs $(r, r+1)$ fall in the range $2^{60} < r < 2^{64}$; 1,064,249 pairs fall in the range $2^{81} < r < 2^{85}$; 31,994 pairs fall in the range $2^{92} < r < 2^{96}$; and, only 1 pair falls in the range $2^{120} < r < 2^{128}$. This pair in the final interval is the largest pair found in this run, with $r = 40120312418488665264241657960477749375$, and factorisations:

$$\begin{aligned} r &= 5^4 \cdot 7 \cdot 13^2 \cdot 17^2 \cdot 19 \cdot 29 \cdot 41 \cdot 109 \cdot 163 \cdot 173 \cdot 239 \cdot 241^2 \cdot 271 \cdot 283 \\ &\qquad \cdot 499 \cdot 509, \text{ and} \\ r+1 &= 2^8 \cdot 3^2 \cdot 31^2 \cdot 43^2 \cdot 47^2 \cdot 83^2 \cdot 103^2 \cdot 311^2 \cdot 479^2 \cdot 523^2. \end{aligned}$$

As we will see later, the number of 64-bit and 85-bit twins we found in this run is enough to find attractive parameters for SQISign. The 96-bit twins will give us parameters with the required smoothness, however we do not have enough pairs to hope to find a prime p where $p^2 - 1$ is divisible by a large power of two.

Table 3 shows that finding many twins of around 128 bits in size is likely to be fruitful in the search for SQISign-friendly parameters, so we ran the algorithm for $B = 1300$ using the `constant-range` optimisation with a range $R = 5000$, in order to specifically target twins $(r, r+1)$ with $r > 2^{115}$. In this run we found 1,091 such pairs - the largest of these pairs is the 145-bit twin $(r, r+1)$ with $r = 36132012096025817587153962195378848686084640$, where

$$\begin{aligned} r &= 2^5 \cdot 5 \cdot 7 \cdot 11^2 \cdot 13 \cdot 23 \cdot 53 \cdot 71 \cdot 109 \cdot 127 \cdot 131 \cdot 193 \cdot 251 \cdot 283 \cdot 307 \\ &\qquad \cdot 359 \cdot 367 \cdot 461 \cdot 613 \cdot 653 \cdot 1277, \text{ and} \\ r+1 &= 3^2 \cdot 29^2 \cdot 31^2 \cdot 43^2 \cdot 59^2 \cdot 61^2 \cdot 73^2 \cdot 79^2 \cdot 89^2 \cdot 167^2 \cdot 401^2 \cdot 419^2. \end{aligned}$$

Among the 1,091 twins CHM found, 184 pairs fall in the range $2^{120} < r < 2^{128}$, which was sufficient to find some SQISign-friendly parameters (though not at all NIST security levels).

In addition, we also ran CHM with $B = 2^{11}$ to obtain a large number of twin smooth integers in the range $2^{55} < r < 2^{100}$ (see Remark 8 in the setting where we want to find twins in such an interval). This run was performed using the `constant-range` optimisation with a range $R = 2500$, and produced 608,233,761 pairs of twins lying in this range. Compared with the $B = 547$ run, the yield from this run gave ample twins with $2^{92} < r < 2^{96}$, which was sufficient to find SQISign parameters with the desirable large power of two.

All of these searches were done using the machine specified in §4.3 - each search took between 1 and 2 days to run.

6.2 Concrete Parameters for SQISign

We now turn to giving a list of SQISign-friendly primes that target NIST Level I, III, and V. Recall from Section 1 that this means that we need to find primes p with $2^f \cdot T \mid p^2 - 1$. We need the exponent f to be as large as possible and the cofactor $T \approx p^{5/4}$ to be B-smooth, aiming to keep the ratio $\sqrt{B}/f$ as small as possible; this quantity is a rough cost metric for the performance of the signing algorithm in SQISign [17, §5.1]. To complement this, the exponent f controls the performance of the verification of SQISign; the larger this exponent is the faster the verification is. We may run into circumstances where the signing cost metric is minimised, but the power of two is not large enough or vice-versa. We aim to balance these as much as possible, thus finding parameters that maximise the power of two while minimising the signing cost metric. We refer to §6.3 for more details on the practicability of our parameters.

Though we need $T \approx p^{5/4}$, if this cofactor is too close to $p^{5/4}$, then the underlying heuristics within the generalised KLPT algorithm might fail and one cannot guarantee a successful signature in SQISign [17, §3.2]. Thus, in practice we need $T \approx p^{5/4+\epsilon}$ for some small ϵ (e.g., $0.02 < \epsilon < 0.1$).

We find parameters for NIST Level I, III and V by searching for 256, 384 and 512-bit primes, respectively. For those primes targeting the higher security levels, these are the first credible SQISign-friendly primes. In what follows, we look at each security level and analyse the most noteworthy primes found in our searches. When stating the factorisations of $p \pm 1$ for the mentioned primes, the underlined factors are the smooth factors of T, while factors in violet are the rough factors which are not needed for SQISign. A full collection of our best SQISign-friendly primes that were found using the CHM machinery is showcased in Table 4.

Remark 9. We note that in all of the forthcoming searches, the post-processing of the CHM twins to find the SQISign-friendly parameters can be made reasonably efficient with straightforward techniques. In particular, the runtime is negligible in comparison to running the CHM searches mentioned in §6.1 and can be done using naive trial division.

NIST I parameters. We targeted 256-bit primes using $n = 2, 3$ and 4. Given that our CHM runs produced a lot more twins of smaller bit-size compared to the 128-bit level, we expect to find more primes using $n = 3, 4$, which was indeed the case. It is worth noting that some primes found with $n = 2$ gave rise to $p^2 - 1$ being divisible by a relatively large power of two. However, in these cases, most of the primes dividing $p^2 - 1$ are relatively large and would therefore give rise to slower isogeny computations during the SQISign protocol [17].

Through the experimentation with the 85-bit twins produced from CHM with $B = 547$, we found the 254-bit prime $p = 2r^3 - 1$ with $r = 20461449125500374748856320$. All the specific criteria that we need for a SQISign parameter set are met, while obtaining an attractively small signing cost metric $\sqrt{B}/f$. For this prime, we have

$$\begin{aligned} p+1 &= 2^{46} \cdot 5^3 \cdot 13^3 \cdot 31^3 \cdot 73^3 \cdot 83^3 \cdot 103^3 \cdot 107^3 \cdot 137^3 \cdot 239^3 \cdot 271^3 \cdot 523^3, \text{ and} \\ p-1 &= 2 \cdot 3^3 \cdot 7 \cdot 11^2 \cdot 17^2 \cdot 19 \cdot 101 \cdot 127 \cdot 149 \cdot 157 \cdot 167 \cdot 173 \cdot 199 \cdot 229 \cdot 337 \\ &\qquad \cdot 457 \cdot 479 \cdot 141067 \cdot 3428098456843 \cdot 4840475945318614791658621. \end{aligned}$$

While the associated cofactor T here exceeds $p^{5/4}$, it does not exceed it by much. As we mentioned earlier, it might therefore be prone to signing failures and hence might not currently be suitable for SQISign. The next 255-bit prime of mention, $p = 2r^3 - 1$ with $r = 26606682403634464748953600$, is very similar to the previous prime, however the cofactor T exceeds $p^{5/4}$ by a larger margin, so would be less prone to these failures. In this case we have

$$\begin{aligned} p+1 &= 2^{40} \cdot 5^6 \cdot 11^3 \cdot 47^3 \cdot 67^6 \cdot 101^3 \cdot 113^3 \cdot 137^3 \cdot 277^3 \cdot 307^3 \cdot 421^3, \text{ and} \\ p-1 &= 2 \cdot 3^2 \cdot 19^3 \cdot 37 \cdot 59 \cdot 61 \cdot 97 \cdot 181^2 \cdot 197 \cdot 223 \cdot 271 \cdot 281 \cdot 311 \cdot 397 \cdot 547 \\ &\qquad \cdot 1015234718965008560203 \cdot 3143438922304814418457. \end{aligned}$$

We additionally ran experiments with the 64-bit twins produced from CHM with $B = 547$ and found a 253-bit prime $p = 2r^4 - 1$ with $r = 8077251317941145600$, where we have

$$\begin{aligned} p+1 &= 2^{49} \cdot 5^8 \cdot 13^4 \cdot 41^4 \cdot 71^4 \cdot 113^4 \cdot 181^4 \cdot 223^4 \cdot 457^4, \text{ and} \\ p-1 &= 2 \cdot 3^2 \cdot 7^5 \cdot 17 \cdot 31 \cdot 53 \cdot 61 \cdot 73 \cdot 83 \cdot 127 \cdot 149 \cdot 233 \cdot 293 \cdot 313 \cdot 347 \cdot 397 \\ &\qquad \cdot 467 \cdot 479 \cdot 991 \cdot 1667 \cdot 19813 \cdot 211229 \cdot 107155419089 \\ &\qquad \cdot 295288804621 \end{aligned}$$

Among all the primes that we found for NIST I security, this appears to be the best. It has both a larger power of two compared to the primes mentioned above found with $n = 3$ and a smaller smoothness bound, thus making the signing cost metric attractively small. Additionally, the cofactor T is large enough to be practical for SQISign without any failures. We note once again that this prime would have been out of scope for the authors of [17] to find since they constrained their search to only find primes for which the power of two is larger than the one found here.

NIST III parameters. We targeted 384-bit primes using $n = 3, 4$ and 6. The challenge in all three of these scenarios is finding enough twins whose product is divisible by a large power of two. With the limited yield of 128-bit twins, finding such primes is not straightforward; the example with $n = 3$ in Table 4 is the only such instance that we managed to find. The picture is somewhat similar with the 96-bit twins: while we have more of them, the success probabilities in Table 3 suggest that we need a lot more twins with a large power of two in order to produce any SQISign-friendly instances. One exceptional prime that was found in this search was the following 375-bit prime, $p = 2r^4 - 1$ with $r = 12326212283367463507272925184$. Here we have

$$\begin{aligned} p+1 &= 2^{77} \cdot 11^4 \cdot 29^4 \cdot 59^4 \cdot 67^4 \cdot 149^4 \cdot 331^4 \cdot 443^4 \cdot 593^4 \cdot 1091^4 \cdot 1319^4, \text{ and} \\ p-1 &= 2 \cdot 3 \cdot 5 \cdot 13 \cdot 17 \cdot 31 \cdot 37 \cdot 53 \cdot 83 \cdot 109 \cdot 131 \cdot 241 \cdot 269 \cdot 277 \cdot 283 \cdot 353 \cdot 419 \\ &\quad \cdot 499 \cdot 661 \cdot 877 \cdot 1877 \cdot 3709 \cdot 9613 \cdot 44017 \cdot 55967 \cdot 522673 \cdot 3881351 \\ &\quad \cdot 4772069 \cdot 13468517 \cdot 689025829 \cdot 30011417945673766253. \end{aligned}$$

Of the NIST Level III primes listed in Table 4, the prime that shows the most promise is the 382-bit prime $p = 2r^6 - 1$ with $r = 11896643388662145024$. Not only is the power of two particularly large but also the smoothness bound of the cofactor T is quite small, reflected in its small signing cost metric (when compared to other p where $p^2 - 1$ is divisible by a large power of 2). We have the factorisations

$$\begin{aligned} p+1 &= 2^{79} \cdot 3^6 \cdot 23^{12} \cdot 107^6 \cdot 127^6 \cdot 307^6 \cdot 401^6 \cdot 547^6, \text{ and} \\ p-1 &= 2 \cdot 5^2 \cdot 7 \cdot 11 \cdot 17 \cdot 19 \cdot 47 \cdot 71 \cdot 79 \cdot 109 \cdot 149 \cdot 229 \cdot 269 \cdot 283 \cdot 349 \cdot 449 \\ &\quad \cdot 463 \cdot 1019 \cdot 1033 \cdot 1657 \cdot 2179 \cdot 2293 \cdot 4099 \cdot 5119 \cdot 10243 \cdot 381343 \\ &\quad \cdot 19115518067 \cdot 7408818089724441233 \cdot 83232143791482135163921. \end{aligned}$$

NIST V parameters. We targeted 512-bit primes using $n = 4$ and 6. Once again, combining our CHM runs with $n = 6$ proved to be the best option for finding SQISign parameters at this level. None of the twins found at the 128-bit level combined with $n = 4$ to produce any SQISign friendly primes. From the set of 85-bit twins found in the $B = 547$ CHM run, the 510-bit prime $p = 2r^6 - 1$ with $r = 31929740427944870006521856$ is particularly attractive. The power of two here is the largest found from this run. Here we have

$$\begin{aligned} p+1 &= 2^{91} \cdot 19^6 \cdot 61^6 \cdot 89^6 \cdot 101^6 \cdot 139^6 \cdot 179^6 \cdot 223^6 \cdot 239^6 \cdot 251^6 \cdot 281^6, \text{ and} \\ p-1 &= 2 \cdot 3^2 \cdot 5 \cdot 7 \cdot 13 \cdot 23 \cdot 29 \cdot 31 \cdot 41 \cdot 53 \cdot 109 \cdot 149 \cdot 157 \cdot 181 \cdot 269 \cdot 317 \cdot 331 \\ &\quad \cdot 463 \cdot 557 \cdot 727 \cdot 10639 \cdot 31123 \cdot 78583 \cdot 399739 \cdot 545371 \cdot 550657 \cdot 4291141 \\ &\quad \cdot 32208313 \cdot 47148917 \cdot 69050951 \cdot 39618707467 \cdot 220678058317 \\ &\quad \cdot 107810984992771213 \cdot 1779937809321608257. \end{aligned}$$

The 85-bit twins found in the CHM run with $B = 2^{11}$ were used to try and find NIST V parameters. The largest power of two that was found in this run which is suitable for SQISign was $f = 109$. The prime with smallest signing cost metric while having a relatively large power of two is the following 508-bit prime, $p = 2r^6 - 1$ where $r =$

26697973900446483680608256. Here we have

$$\begin{aligned}p+1 &= 2^{85}\cdot 17^{12}\cdot 37^{6}\cdot 59^{6}\cdot 97^{6}\cdot 233^{6}\cdot 311^{12}\cdot 911^{6}\cdot 1297^{6},\ \text{and}\\ p-1 &= 2\cdot 3^{2}\cdot 5\cdot 7\cdot 11^{2}\cdot 23^{2}\cdot 29\cdot 127\cdot 163\cdot 173\cdot 191\cdot 193\cdot 211\cdot 277\cdot 347\cdot 617\\ &\quad\cdot 661\cdot 761\cdot 1039\cdot 4637\cdot 5821\cdot 15649\cdot 19139\cdot 143443\cdot 150151\cdot 3813769\\ &\quad\cdot 358244059\cdot 992456937347\cdot 35324048178196536982389750 7\\ &\quad\cdot 86010200695145744013716588914030 21.\end{aligned}$$

6.3 Performance Estimates

We would ideally implement our primes using the SQISign code provided in [17] to determine how well these parameters perform in practice. However, the current implementation is specifically tailored towards the particular primes that are being used, and is limited to NIST I parameter sizes. Including our NIST I primes from Table 4 results in failures during key generation, which seem to stem from using parameters with different powers of two. Thus, implementing and benchmarking our parameters would require a major rework of the provided code, which is out of the scope of this work.

NIST I. The state-of-the-art implementation of SQISign uses a 254-bit prime that was found using the extended Euclidean algorithm (XGCD) [9,16] (see §2). With this method, it is possible to, for example, force $p \pm 1$ and $p \mp 1$ to be divisible by a large power of 2 and 3 (respectively). Indeed, with this approach, a smooth factor of size $\approx \sqrt{p}$ comes for free in both $p \pm 1$.

Concretely, the prime p_{3923} used in [17] has

$$\begin{aligned}p+1 &= 2^{65}\cdot 5^{2}\cdot 7\cdot 11\cdot 19\cdot 29^{2}\cdot 37^{2}\cdot 47\cdot 197\cdot 263\cdot 281\cdot 461\cdot 521\cdot 3923\cdot 62731\\ &\quad\cdot 96362257\cdot 3924006112952623,\ \text{and}\\ p-1 &= 2\cdot 3^{65}\cdot 13\cdot 17\cdot 43\cdot 79\cdot 157\cdot 239\cdot 271\cdot 283\cdot 307\cdot 563\cdot 599\cdot 607\cdot 619\\ &\quad\cdot 743\cdot 827\cdot 941\cdot 2357\cdot 10069.\end{aligned}$$

The primes from Table 4 provide various alternatives for NIST I parameters, and we can give simplified estimates for their performance in comparison to p_{3923}. As an example, we will consider p_{479}, the 253-bit prime from Table 4 having $B = 479$. With $f = 49$, it features a slightly smaller power of two compared to p_{3923} with $f = 65$. This means that we would have to verify the signature isogeny in 21 chunks of 2^{49}-isogenies, instead of 16 chunks of 2^{65}-isogenies for p_{3923}. Given that the computational bottleneck for this is the generation of the respective kernel points per chunk, and ignoring the savings of computing 2^{49}-isogenies instead of 2^{65}-isogenies and the relatively cheap recomputation of the challenge isogeny, this results in an estimated slowdown of roughly $21/16 \approx 1.31$.

NIST security level	p		$\lceil \log_2(p) \rceil$	f	B	$\sqrt{B}/f$	$\log_p(T)$
NIST I	p_{3923}[17]		254	65	3923	0.96	1.32
	n	r					
	2	1211460311716772790566574529001291776	241	49	1091	0.67	1.28
		2091023014142971802357816084152713216	243	49	887	0.61	1.28
	3	3474272816789867297357824	246	43	547	0.54	1.29
		10227318375788227199589376	251	31	383	0.63	1.31
		21611736033260878876800000	254	31	421	0.66	1.28
		20461449125500374748856320	254	46	523	0.50	1.26
		26606682403634464748953600	255	40	547	0.58	1.28
	4	1466873880764125184	243	49	701	0.54	1.28
		8077251317941145600	253	49	479	0.45	1.30
		12105439990105079808[17]	255	61	1877	0.71	1.31
		13470906659953016832[17]	256	61	1487	0.63	1.30
NIST III	3	1374002035005713149550405343373848576	362	37	1277	0.97	1.25
	4	5139734876262390964070873088	370	45	11789	2.41	1.26
		12326212283367463507272925184	375	77	55967	3.07	1.31
		18080754980295452456023326720	377	61	95569	5.07	1.26
		27464400309146790228660255744	379	41	13127	2.79	1.29
	6	2628583629218279424	369	73	13219	1.58	1.27
		5417690118774595584	375	79	58153	3.05	1.27
		11896643388662145024	382	79	10243	1.28	1.30
	12	5114946480[13]	389	49	31327	3.61	1.30
NIST V	6	9469787780580604464332800	499	109	703981	7.70	1.25
		12233468605740686007808000	502	73	376963	8.41	1.28
		26697973900446483680608256	508	85	150151	4.56	1.26
		31929740427944870006521856	510	91	550657	8.15	1.25
		41340248200900819056793600	512	67	224911	7.08	1.28

Table 4: A table of SQISign parameters $p = p_n(r)$ for twin-smooth integers $(r, r \pm 1)$ found using CHM at each security level. The f is the power of two dividing $(p^2-1)/2$ and B is the smoothness bound of the odd cofactor $T \approx p^{5/4}$. It also includes existing primes in the literature including the state-of-the-art.

Thus, we expect a modest slowdown from a verification time of 6.7ms (see [17]) to roughly 8.8ms on a modern CPU.

However, we expect a significant speedup for signing: The computational bottleneck during the signature generation is the repeated computation of T-isogenies; one computes two T isogenies per chunk of 2^f-isogenies in the verification. Since the T-isogeny computation is dominated by its largest prime factor B, and its cost can be estimated by $\sqrt{B}$, the ratio of the signing cost metrics $\sqrt{B}/f$ from Table 4 reflects the overall comparison. Given this metric, we expect a speedup factor of roughly $0.45/0.96 \approx 0.47$. For the running time, this would mean an improvement from 424ms (see [17]) to roughly 199ms on a modern CPU.

We can also consider a different cost-estimate, given by summing the cost $\sqrt{\ell_i}$ for the five biggest (not necessarily distinct) prime factors $\ell_i \mid T$, before dividing by f. The advantage of considering more factors of T is that it constitutes a larger portion of the time it takes to compute a T-isogeny, while the disadvantage is that the cost $\sqrt{\ell}$ becomes increasingly inaccurate for smaller prime factors ℓ. In this metric, the speedup is smaller, but is still significant. Specifically, we expect a speedup factor of roughly $2.19/3.04 \approx 0.72$, which would result in an improvement from 424ms to roughly 305ms.

In a nutshell, even though we can only give rough estimates for running times, we expect our NIST I parameters to achieve much better signing times due to the smaller smoothness bounds B, at the cost of a very modest slowdown for verification due to slightly smaller values of f. In the light of the relatively slow signing times in SQISign, this option seems worthwhile for applications that require faster signing.

NIST III and V. As mentioned earlier, our work showcases the first credible primes for SQISign at the NIST III and NIST V security level. A beneficial feature about most of the primes found in Table 4 is that the majority of the smooth factors are relatively small (e.g. $B < 2^{10}$). In comparison, we expect the XGCD method to scale worse for larger security levels, i.e., requiring much larger smoothness bounds. This is similar to the analysis in [10], which shows that while the XGCD approach has reasonable smoothness probabilities for NIST I parameters, other methods become superior for larger sizes.

We note that there are other 384 and 512-bit primes in the literature for which $p^2 - 1$ is smooth [10,13]. None of the primes from [10] have a large enough power of two for a suitable SQISign application. Some primes were found in the context of the isogeny-based public-key encryption scheme Séta [13] that could be suitable for SQISign. As part of their parameter setup, they required finding $\approx$ 384-bit primes[10]. Of the 7 primes that they found, the 389-bit prime, $p = 2r^{12} - 1$ with $r = 5114946480$ appears to be somewhat SQISign-friendly to achieve NIST III security (see Table 4). However, in addition to its worse signing metric, representations of $\mathbb{F}_p$-values require an additional register in

[10] That satisfy some mild conditions outside of just requiring $p^2 - 1$ to be smooth

this case compared to our primes of bitlengths slightly below 384. Thus, we can expect implementations of $\mathbb{F}_p$-arithmetic to perform significantly worse for this prime.

Remark 10. The requirement we impose on $p^2 - 1$ being divisible by $2^f \cdot T$ is to ensure that it fits within the current implementation of SQISign. At present, the SQISign implementation has a fine-grained optimisation of their ideal to isogeny algorithm to the setting with $\ell = 2$. In general, one could instead allow $p^2 - 1$ to be divisible by $L \cdot T$, for a smooth number L with $\gcd(L, T) = 1$. This could open new avenues to find SQISign-friendly primes, but would require a reconfiguration of the SQISign code. For example, using the prime found with $r = 2091023014142971802357816084152713216$ from Table 4, we could use $L = 2^{49} \cdot 3^4 \cdot 5 \mid p^2 - 1$ and still have a large enough smooth factor T to exceed $p^{5/4}$, thereby further minimising the expected slowdown for verification.

Remark 11. The focus of this work has been on finding parameters for SQISign but there are other isogeny-based cryptosystems that could benefit from such quadratic twist-style primes. While traditional SIDH [19] is now broken, there have been proposed countermeasures [18,3,2] that aim to thwart the attacks from [6,22,23]. Currently, these countermeasures use SIDH-style primes, but could potentially benefit from quadratic twist-style primes like those explored in this work for SQISign. However, these countermeasure require primes of larger sizes, so it is unclear if our CHM-based approach scales to these sizes, especially when aiming to balance the size of the smooth cofactors of $p + 1$ and $p - 1$. Nevertheless, our techniques might give a good starting point for future research in this direction.

6.4 Other Techniques for Finding SQISign Parameters

As seen in §2, we can collect twin smooth integers via different methods, and use them as inputs to $p_n(x)$ to search for primes. Though these alternative methods are expected to have greater smoothness bound, they have certain advantages. Namely, we are able to force larger powers of 2 into $p^2 - 1$ and search for twin smooths of large bitsizes (targeting NIST-III and -V).

Although we expect most primes in this section to perform worse when instantiated in SQISign compared to the primes from §6.2, their concrete performance cannot be evaluated with the software from [16,17] (see §6.3). In this section, we present the best primes found with each approach in the hopes that future implementations of SQISign benefit from a larger pool of potential primes to choose from. We give a list of these primes in Table 5

XGCD twin smooths. For generating smaller twins, the XGCD approach can be used to yield relatively high smoothness probabilities. Although this increases the smoothness bound compared to CHM, we can choose smooth factors of roughly n bits combined when searching for n-bit twin smooths. This allows us to force larger powers of 2.

As an example, the 261-bit prime $p = 2r^4 - 1$ with $r = 348482182313552117 76$ was found using this approach. Here we have

$$\begin{aligned} p + 1 &= 2^{77} \cdot 3^{20} \cdot 23^4 \cdot 151^4 \cdot 157^4 \cdot 233^4 \cdot 2153^4, \text{ and} \\ p - 1 &= 2 \cdot 5^2 \cdot 17 \cdot 41 \cdot 61 \cdot 71 \cdot 97 \cdot 101^2 \cdot 113 \cdot 137 \cdot 257 \cdot 263 \cdot 313 \cdot 353 \cdot 547 \cdot 853 \\ &\quad \cdot 1549 \cdot 2017 \cdot 2081 \cdot 2311 \cdot 3019 \cdot 24989 \cdot 58601 \cdot 5511340166779281313. \end{aligned}$$

This prime is similar to the primes found in [17], giving a smaller smoothness bound and a larger power of 2 compared to the state-of-the-art. However, it exceeds the size of 256 bits, and thus we expect it to perform significantly worse due to the fact that representations of values in $\mathbb{F}_p$ require an additional register in this case. Additionally, a large majority of the factors in $p^2 - 1$ are relatively large, making isogeny computations rather slow. This is consistent with the primes in [17].

PTE twin smooths. As the number of 128-bit twins that were found using CHM is relatively small, in some cases we were not able to find suitable SQISign parameters. This mainly concerns the setting using $n = 4$ and finding NIST-V parameters, for which data from the CHM run with $B < 1300$ did not yield any NIST-V SQISign-friendly instances.

To find more large twins, we can use the PTE approach [10] (see §2) to find 2^{14}-smooth 128-bit twins, sacrificing the smaller smoothness bounds that were used during our CHM runs. In total, we found 3,648 such 128-bit twins that resulted in a prime of the form $p = 2r^4 - 1$. Of these, two primes show strong potential to be used in SQISign and are thus also given in Table 4.

Larger values of n. We could also consider finding primes of the form $p = 2r^n - 1$ for larger values of n, where the only restriction is that r is a smooth number. Compared to the previous ideas this restriction decreases the amount of guaranteed smoothness, but if n is chosen carefully then we can obtain increased smoothness probabilities. The polynomial $p_n(x)^2 - 1$ is highly related to the cyclotomic polynomials Φ_d for $d \mid n$ as

$$p_n(x)^2 - 1 = 2x^n(2x^n - 2) = 4x^n(x^n - 1) = 4x^n \prod_{d|n} \Phi_d.$$

Recall that Φ_d is an irreducible polynomial of degree $\varphi(d)$, where φ denotes Euler's totient function. Therefore, the largest irreducible factor of $p_n(x)^2 - 1$ is of degree $\varphi(n)$. This in turn means that the largest factor that $p = 2r^n - 1$ can possibly have is around the size of $r^{\varphi(n)} \approx p^{\varphi(n)/n}$. Therefore, we would like to minimise the value $\varphi(n)/n$.

As we allow n to increase, this value can get arbitrarily low. Indeed, setting $n = P_k$, where P_k denotes the k-th primorial, we find that

$$\frac{\varphi(P_k)}{P_k} = \prod_{i=1}^{k} \frac{p_i - 1}{p_i} = \prod_{i=1}^{k} \left(1 - \frac{1}{p_i}\right),$$

and as k goes towards infinity, we see that

$$\lim_{k \to \infty} \frac{\varphi(P_k)}{P_k} = \prod_{i=1}^{\infty} \left(1 - \frac{1}{p_i}\right) = \frac{1}{\zeta(1)},$$

where $\zeta(s)$ denotes the Riemann-zeta function, which has a pole at $s = 1$.

However, we cannot allow n to become too large; we still need a sufficiently large range of inputs, so that there exists a smooth r such that $2r^n - 1$ is prime. Therefore, consider a bound $n < B_n$ where B_n is chosen such that we can still have a large search space. Based on the multiplicative property of the totient function, the fact that $\varphi(q) = q - 1$ when q is prime, and the fact that

$$\frac{\varphi(n)}{n} = \frac{\varphi(\text{rad}(n))}{\text{rad}(n)},$$

where $\text{rad}(n)$ denotes the square-free part of n, the optimal choices of n are in the set

$$n \in \{2^{e_1} 3^{e_2} \dots p_k^{e_k} < B_n \mid P_k < B_n < P_{k+1}, e_i \geq 1\},$$

where P_k again denotes the k-th primorial.

As an example, we look for NIST-V parameters $p \in [2^{500}, 2^{512}]$. If we want at least a range of size 2^{25} such that $2r^n - 1 \in [2^{500}, 2^{512}]$, we see that we have to have $n < B_n = 20$. Therefore, our set of optimal choices of n becomes

$$n \in \{2 \cdot 3, 2^2 \cdot 3, 2 \cdot 3^2\} = \{6, 12, 18\}.$$

Using $n = 6$, the range of suitable r-values becomes large enough that we cannot search through all of them. Thus, searches would require further restrictions on the suitable r-values, such as only considering twin-smooths.

For $n \in \{12, 18\}$, we can exhaust the full search space, and obtain several promising candidates. These are include in Table 4. Among all of these, the 510-bit prime $p = 2r^{12} - 1$ with $r = 5594556480768$ seems very suitable for NIST-V. It has a low cost factor and has a large power of three, which could be beneficial for SQISign implementations. Here we have

$$\begin{aligned}
p + 1 &= 2^{97} \cdot 3^{60} \cdot 239^{12} \cdot 571^{12} \cdot 659^{12}, \text{ and} \\
p - 1 &= 2 \cdot 5^2 \cdot 7 \cdot 13^2 \cdot 17 \cdot 19 \cdot 43 \cdot 83 \cdot 103 \cdot 109 \cdot 139^2 \cdot 151 \cdot 223 \cdot 277 \cdot 317 \cdot 1249 \\
&\quad \cdot 1373 \cdot 2311 \cdot 3067 \cdot 4133 \cdot 28279 \cdot 28447 \cdot 40087 \cdot 60089 \cdot 69073 \cdot 88469 \\
&\quad \cdot 2226517 \cdot 5856073 \cdot 6242671 \cdot 14237127193 \cdot 25752311173 \\
&\quad \cdot 63101553683977 \cdot 383802498444339986625038 41.
\end{aligned}$$

NIST security level	n	r	$\lceil \log_2(p) \rceil$	f	B	$\sqrt{B}/f$	$\log_p(T)$
NIST-I	4	34848218231355211776	261	77	2311	0.62	1.30
NIST-III	12	2446635904	376	85	9187	1.13	1.29
NIST-V	4	114216781548581709439512875801279791104	507	65	75941	4.24	1.26
		123794274387474298912742543819242587136	508	41	15263	3.01	1.29
	12	5594556480768	510	97	88469	3.07	1.29
	18	335835120	511	73	24229	2.13	1.29

Table 5: A table of SQISign parameters $p = p_n(r)$ found using twin-smooth integers $(r, r \pm 1)$ at each security level. The twins used here were not found using CHM. The other quantities are just as in Table 4.

Remark 12. Unlike the CHM method and similar methods, we cannot generate more values to input into this technique, as the amount is small enough to quickly exhaust the full search space. This is in stark contrast to CHM, which could potentially - given more computing power - generate more twin smooths of given sizes to give new suitable SQISign parameters. Hence, we conclude that the CHM method with smaller values of n will ultimately give rise to new, better SQISign parameters than the ones found with the higher values of n.

Acknowledgements. We thank Joost Renes for several helpful discussions about the CHM algorithm, and Luca De Feo for helpful comments on SQISign prime requirements during the preparation of this work as well as the anonymous reviewers for their constructive feedback.

References

1. W. D. Banks and I. E. Shparlinski. Integers with a large smooth divisor. *arXiv preprint math/0601460*, 2006.
2. A. Basso and T. B. Fouotsa. New sidh countermeasures for a more efficient key exchange. Cryptology ePrint Archive, Paper 2023/791, 2023. https://eprint.iacr.org/2023/791.
3. A. Basso, L. Maino, and G. Pope. FESTA: Fast encryption from supersingular torsion attacks. Cryptology ePrint Archive, Paper 2023/660, 2023. https://eprint.iacr.org/2023/660.
4. D. J. Bernstein, L. De Feo, A. Leroux, and B. Smith. Faster computation of isogenies of large prime degree. *Open Book Series*, 4(1):39–55, 2020.
5. T. Bingmann. TLX: Collection of sophisticated C++ data structures, algorithms, and miscellaneous helpers, 2018. https://panthema.net/tlx, retrieved Oct. 7, 2020.
6. W. Castryck and T. Decru. An efficient key recovery attack on SIDH. In *EUROCRYPT*, volume 14008 of *Lecture Notes in Computer Science*, pages 423–447. Springer, 2023.

7. J. B. Conrey and M. A. Holmstrom. Smooth values of quadratic polynomials. *Experimental Mathematics*, 30(4):447–452, 2021.
8. J. B. Conrey, M. A. Holmstrom, and T. L. McLaughlin. Smooth neighbors. *Experimental Mathematics*, 22(2):195–202, 2013.
9. C. Costello. B-SIDH: supersingular isogeny Diffie-Hellman using twisted torsion. In *ASIACRYPT*, volume 12492 of *Lecture Notes in Computer Science*, pages 440–463. Springer, 2020.
10. C. Costello, M. Meyer, and M. Naehrig. Sieving for twin smooth integers with solutions to the Prouhet-Tarry-Escott problem. In *EUROCRYPT*, volume 12696 of *Lecture Notes in Computer Science*, pages 272–301. Springer, 2021.
11. P. Dartois, A. Leroux, D. Robert, and B. Wesolowski. SQISignHD: New dimensions in cryptography. Cryptology ePrint Archive, Paper 2023/436, 2023. https://eprint.iacr.org/2023/436.
12. N. G. de Bruijn. On the number of positive integers $\leq$ x and free of prime factors $> y$, ii. *Indag. Math*, 38:239–247, 1966.
13. L. De Feo, C. Delpech de Saint Guilhem, T. B. Fouotsa, P. Kutas, A. Leroux, C. Petit, J. Silva, and B. Wesolowski. Séta: Supersingular encryption from torsion attacks. In *ASIACRYPT*, volume 13093 of *Lecture Notes in Computer Science*, pages 249–278. Springer, 2021.
14. K. Dickman. On the frequency of numbers containing prime factors of a certain relative magnitude. *Arkiv for matematik, astronomi och fysik*, 22(10):A–10, 1930.
15. J. Komada Eriksen, L. Panny, J. Sotáková, and M. Veroni. Deuring for the people: Supersingular elliptic curves with prescribed endomorphism ring in general characteristic. *Cryptology ePrint Archive*, 2023.
16. L. De Feo, D. Kohel, A. Leroux, C. Petit, and B. Wesolowski. SQISign: compact post-quantum signatures from quaternions and isogenies. In *ASIACRYPT*, volume 12491 of *Lecture Notes in Computer Science*, pages 64–93. Springer, 2020.
17. L. De Feo, A. Leroux, P. Longa, and B. Wesolowski. New algorithms for the deuring correspondence - towards practical and secure sqisign signatures. In *EUROCRYPT*, volume 14008 of *Lecture Notes in Computer Science*, pages 659–690. Springer, 2023.
18. T. B. Fouotsa, T. Moriya, and C. Petit. M-SIDH and MD-SIDH: Countering sidh attacks by masking information. In *EUROCRYPT*, volume 14008 of *Lecture Notes in Computer Science*, pages 282–309. Springer, 2023.
19. D. Jao and L. De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *PQCrypto*, volume 7071 of *Lecture Notes in Computer Science*, pages 19–34. Springer, 2011.
20. D. H. Lehmer. On a problem of Störmer. *Illinois Journal of Mathematics*, 8(1):57–79, 1964.
21. F. Luca and F. Najman. On the largest prime factor of x^2-1. *Mathematics of computation*, 80(273):429–435, 2011.
22. L. Maino, C. Martindale, L. Panny, G. Pope, and B. Wesolowski. A direct key recovery attack on SIDH. In *EUROCRYPT*, volume 14008 of *Lecture Notes in Computer Science*, pages 448–471. Springer, 2023.
23. D. Robert. Breaking SIDH in polynomial time. In *EUROCRYPT*, volume 14008 of *Lecture Notes in Computer Science*, pages 472–503. Springer, 2023.
24. C. Størmer. Quelques théorèmes sur l'équation de Pell $x^2 - dy^2 = \pm 1$ et leurs applications. *Christiania Videnskabens Selskabs Skrifter, Math. Nat. Kl*, (2):48, 1897.

25. G. Tenenbaum. Integers with a large friable component. *Acta arithmetica*, 124:287–291, 2006.
26. G. Tenenbaum. *Introduction to analytic and probabilistic number theory*, volume 163. American Mathematical Soc., 2015.
27. The National Institute of Standards and Technology (NIST). Submission requirements and evaluation criteria for the post-quantum cryptography standardization process, December, 2016.
28. The National Institute of Standards and Technology (NIST). Call for additional digital signature schemes for the post-quantum cryptography standardization process, October, 2022.

AprèsSQI: Extra Fast Verification for SQIsign Using Extension-Field Signing

Maria Corte-Real Santos, Jonathan Komada Eriksen, Michael Meyer, and Krijn Reijnders

Accepted at EuroCrypt 2024 (best young research paper). Full version available at: eprint.iacr.org/2023/1559.pdf

AprèsSQI: Extra Fast Verification for SQIsign Using Extension-Field Signing

Maria Corte-Real Santos[1], Jonathan Komada Eriksen[2], Michael Meyer[3], and Krijn Reijnders[4]

[1] University College London
maria.santos.20@ucl.ac.uk
[2] Norwegian University of Science and Technology
jonathan.k.eriksen@ntnu.no
[3] University of Regensburg, Germany
michael@random-oracles.org
[4] Radboud University, Nijmegen, The Netherlands
krijn@cs.ru.nl

Abstract. We optimise the verification of the SQIsign signature scheme. By using field extensions in the signing procedure, we are able to significantly increase the amount of available rational 2-power torsion in verification, which achieves a significant speed-up. This, moreover, allows several other speed-ups on the level of curve arithmetic. We show that the synergy between these high-level and low-level improvements gives significant improvements, making verification 2.07 times faster, or up to 3.41 times when using size-speed trade-offs, compared to the state of the art, without majorly degrading the performance of signing.

Keywords: post-quantum cryptography, isogenies, SQIsign, verification

1 Introduction

Research has shown that large-scale quantum computers will break current public-key cryptography, such as RSA or ECC, whose security relies on the hardness of integer factorization or the discrete logarithm, respectively [36]. Post-quantum cryptography seeks to thwart the threat of quantum computers by developing cryptographic primitives based on alternative mathematical problems that cannot be solved efficiently by quantum computers. In recent years, lattice-based cryptography has developed successful post-quantum

Work by the researchers formerly known as SICQ.
Author list in alphabetical order; see https://www.ams.org/profession/leaders/CultureStatement04.pdf. This work has been supported by UK EPSRC grant EP/S022503/1 and by the German Federal Ministry of Education and Research (BMBF) under the project 6G-RIC (ID 16KISK033).

schemes for essential primitives such as key encapsulation mechanisms (KEMs) and digital signatures that will be standardized by NIST. Lattice-based signatures are able to provide fast signing and verification, but have to resort to larger key and signature sizes than were previously acceptable in pre-quantum signatures. For applications where the amount of data transmitted is crucial, these lattice-based schemes may not be a practical option. NIST is therefore looking for other digital signature schemes with properties such as smaller combined public key and signature sizes to ensure a smooth transition to a post-quantum world [39].

A potential solution to this problem is provided by the sole isogeny-based candidate in NIST's new call for signatures – SQIsign [21] – as it is currently the candidate that comes closest to the data sizes transmitted (i.e. the combined size of the signature and the public key) in pre-quantum elliptic curve signatures [28, 29]. SQIsign is most interesting in scenarios that require small signature sizes and fast verification, particularly in those applications where the performance of signing is not the main concern. A few common examples include long-term signatures, specifically public-key certificates, code updates for small devices, authenticated communication with embedded devices or other microcontrollers that solely run verification, and smart cards. For such use cases it is imperative to bring down the cost of verification as much as possible.

Performance bottlenecks in SQIsign. The bottleneck of verification in SQIsign is the computation of an isogeny of fixed degree 2^e with $e \approx (15/4)\log(p)$, where p denotes the prime one is working over, e.g. $\log(p) \approx 256$ for NIST Level I security. However, the rational 2-power torsion, from here on denoted as the $2^\bullet$-torsion, is limited, since we work with supersingular elliptic curves over $\mathbb{F}_{p^2}$ of order $(p+1)^2$ and $(p-1)^2$. This sets a theoretical limit of $2^{\log p}$ for the $2^\bullet$-torsion. Therefore, the verifier needs to perform several *blocks* of degree $2^\bullet$ to complete the full 2^e-isogeny, where each of these blocks involves costly steps such as computing a $2^\bullet$-torsion basis or isogeny kernel generator. Hence, in general, a smaller number of blocks improves the performance of verification.

On the other hand, the bottleneck in signing is the computation of several T-isogenies for odd smooth $T \approx p^{5/4}$. Current implementations of SQIsign therefore require $T \mid (p-1)(p+1)$, such that $\mathbb{F}_{p^2}$-rational points are available for efficient T-isogeny computations. The performance of this step is dominated by the smoothness of T, i.e., its largest prime factor.

While this additional divisibility requirement theoretically limits the maximal $2^\bullet$-torsion to roughly $p^{3/4}$, current techniques for finding SQIsign-friendly primes suggest that achieving this with acceptable smoothness of T is infeasible [10, 12, 14, 18, 21]. In particular, the NIST submission of SQIsign uses a prime with rational 2^{75}-torsion and 1973 as largest factor of T. Since $e \approx (15/4) \cdot 256 = 960$, this means that the verifier has to perform $\lceil e/75 \rceil = 13$ costly isogeny blocks. Increasing the $2^\bullet$-torsion further is difficult as it decreases the probability of finding a smooth and large enough T for current implementations of SQIsign.

Our contributions. In this work, we deploy a range of techniques to increase the $2^\bullet$-torsion and push the SQIsign verification cost far below the state of the art. Alongside these technical contributions, we aim to give an accessible description of SQIsign, focusing primarily on verification, which solely uses elliptic curves and isogenies and does not require knowledge of quaternion algebras.

Even though we target faster verification, our main contribution is signing with field extensions. From this, we get a much weaker requirement on the prime p, which in turn enables us to increase the size of the $2^\bullet$-torsion.

Focusing on NIST Level I security, we study the range of possible $2^\bullet$-torsion to its theoretical maximum, and measure how its size correlates to verification time through an implementation that uses an equivalent to the number of field multiplications as cost metric. Compared to the state of the art, increasing the $2^\bullet$-torsion alone makes verification almost 1.7 times faster. Further, we implement the new signing procedure as proof-of-concept in SageMath and show that signing times when signing with field extensions are in the same order of magnitude as when signing only using operations in $\mathbb{F}_{p^2}$.

For verification, in addition to implementing some known general techniques for improvements compared to the reference implementation provided in the NIST submission of SQIsign, we show that increasing the $2^\bullet$-torsion also opens up a range of optimisations that were previously not possible. For instance, large $2^\bullet$-torsion allows for an improved challenge-isogeny computation and improved basis and kernel generation. Furthermore, we show that size-speed trade-offs as first proposed by De Feo, Kohel, Leroux, Petit, and Wesolowski [21] become especially worthwhile for large $2^\bullet$-torsion. When pushing the $2^\bullet$-torsion to its theoretical maximum, this even allows for uncompressed signatures, leading to significant speed-ups at the cost of roughly doubling the signature sizes.

For two specific primes with varying values of $2^\bullet$-torsion, we combine all these speed-ups, and measure the performance of verification. Compared to the implementation of the SQIsign NIST submission [12], we reach a speed-up up to a factor 2.70 at NIST Level I when keeping the signature size of 177 bytes. When using our size-speed trade-offs, we reach a speed-up by a factor 3.11 for signatures of 187 bytes, or a factor 4.46 for uncompressed signatures of 322 bytes. Compared to the state of the art [31], these speed-ups are factors 2.07, 2.38 and 3.41 respectively.

Related work. De Feo, Kohel, Leroux, Petit, and Wesolowski [21] published the first SQIsign implementation, superseded by De Feo, Leroux, Longa, and Wesolowski [22]. Subsequently, Lin, Wang, Xu, and Zhao [31] introduced several improvements for this implementation. The NIST submission of SQIsign [12] features a new implementation that does not rely on any external libraries. Since this is the latest and best documented implementation, we will use it as a baseline for performance comparison, and refer to it as SQIsign (NIST). Since the implementation by Lin, Wang, Xu, and Zhao [31] is not publicly available, we included their main improvement for verification in SQIsign (NIST), and refer to this as SQIsign (LWXZ).

Dartois, Leroux, Robert, and Wesolowski [19] recently introduced SQIsignHD, which massively improves the signing time in SQIsign, in addition to a number of other benefits, but at the cost of a still unknown slowdown in verification. This could make SQIsignHD an interesting candidate for applications that prioritise the combined cost of signing and verification over the sole cost of verification.

Recent work by Eriksen, Panny, Sotáková, and Veroni [24] explored the feasibility of computing the Deuring correspondence (see Section 2.2) for *general* primes p via using higher extension fields. We apply the same techniques and tailor them to *specialised* primes for use in the signing procedure of SQIsign.

Overview. The rest of the paper is organised as follows. Section 2 recalls the necessary background, including a high-level overview of SQIsign. Section 3 describes how using field extensions in signing affects the cost and relaxes requirements on the prime. Section 4 analyses how the size of the $2^\bullet$-torsion correlates to verification time. Section 5 presents optimisations enabled by the increased $2^\bullet$-torsion, while Section 6 gives further optimisations enabled by increased signature sizes. Finally, Section 7 gives some example parameters, and measures their performance compared to the state of the art.

Availability of software. We make our Python and SageMath software publically available under the MIT licence at

https://github.com/TheSICQ/ApresSQI.

2 Preliminaries

Throughout this paper, p denotes a prime number and $\mathbb{F}_{p^k}$ the finite field with p^k elements, where $k \in \mathbb{Z}_{>0}$.

2.1 Elliptic curves and their endomorphism rings.

We first give the necessary geometric background to understand the SQIsign signature scheme. For a more general exposition we refer to Silverman [38].

Isogenies. An isogeny $\varphi : E_1 \to E_2$ between two elliptic curves E_1, E_2 is a non-constant morphism that sends the identity of E_1 to the identity of E_2. The degree $d = \deg(\varphi)$ of an isogeny is its degree as a rational map. If the degree d of an isogeny φ has the prime factorisation $d = \prod_{i=1}^{n} \ell_i^{e_i}$, we can decompose φ into the composition of e_i isogenies of degree ℓ_i for $i = 1$ to n. For every isogeny $\varphi : E_1 \to E_2$, there is a (unique) *dual* isogeny $\widehat{\varphi} : E_2 \to E_1$ that satisfies $\widehat{\varphi} \circ \varphi = [\deg(\varphi)]$, the multiplication-by-deg(φ) map on E_1. Similarly, $\varphi \circ \widehat{\varphi}$ is the multiplication by $\deg(\varphi)$ on E_2.

A *separable* isogeny is described, up to isomorphism, by its kernel, a group of order d. Given a kernel G of prime order d, we can compute the corresponding isogeny $\phi : E \to E/G$ using Vélu's formulas [41] in $\widetilde{O}(d)$. Bernstein, De Feo, Leroux, and Smith [8] showed that this can be asymptotically reduced to $\widetilde{O}(\sqrt{d})$ using $\sqrt{\text{élu}}$ formulas. In Section 2.5, we return to the topic of computing isogenies and give a more detailed discussion.

Endomorphism rings. An isogeny from a curve E to itself is called an *endomorphism*. For example, for any integer n, the multiplication-by-n map is an endomorphism. Another, not necessarily distinct, example for elliptic curves defined over $\mathbb{F}_q$ is the *Frobenius endomorphism* $\pi : (x, y) \mapsto (x^q, y^q)$.

The set of endomorphisms $\mathrm{End}(E)$ of an elliptic curve E forms a ring under (pointwise) addition and composition of isogenies. The endomorphism ring of $E/\overline{\mathbb{F}}_p$ is either isomorphic to an imaginary quadratic order, or to a maximal order in a quaternion algebra ramified at p and ∞ (which will be defined in Section 2.2). In the latter case, we say that E is *supersingular*, and from this point forward, E will denote a supersingular curve.

Supersingular elliptic curves and their isomorphism classes. We will mostly consider supersingular elliptic curves *up to isomorphism*, and thus work with isomorphism classes of these curves. Throughout, we will exploit the fact that every isomorphism class of supersingular curves has a model over $\mathbb{F}_{p^2}$, such that the p^2-power Frobenius π is equal to the multiplication-by-$(-p)$ map. Such curves E are $\mathbb{F}_{p^2}$-isogenous to curves defined over $\mathbb{F}_p$, and satisfy

$$E(\mathbb{F}_{p^{2k}}) = E\left[p^k - (-1)^k\right] \cong \mathbb{Z}/\left(p^k - (-1)^k\right)\mathbb{Z} \oplus \mathbb{Z}/\left(p^k - (-1)^k\right)\mathbb{Z}, \tag{1}$$

while their quadratic twist over $\mathbb{F}_{p^{2k}}$, which we will denote E_k^t, satisfies

$$E_k^t(\mathbb{F}_{p^{2k}}) = E\left[p^k + (-1)^k\right] \cong \mathbb{Z}/\left(p^k + (-1)^k\right)\mathbb{Z} \oplus \mathbb{Z}/\left(p^k + (-1)^k\right)\mathbb{Z}. \tag{2}$$

For such curves, for any positive integer $N \mid p^k \pm 1$, the full N-torsion group $E[N]$ is defined over $\mathbb{F}_{p^{2k}}$, either on the curve itself, or on its twist.

The isogeny problem. The fundamental hard problem underlying the security of all isogeny-based primitives is the following: given two elliptic curves E_1, E_2 defined over $\mathbb{F}_{p^2}$ find an isogeny $\phi : E_1 \to E_2$. The best classical attack against this problem is due to Delfs and Galbraith [23] which runs in time $\widetilde{O}(\sqrt{p})$, and quantum attack due to Biasse, Jao, and Sankar [9] that runs in $\widetilde{O}(\sqrt[4]{p})$. A related problem, which will be useful in the context of SQIsign, is the *endomorphism ring problem*, which asks, given a supersingular elliptic curve $E/\mathbb{F}_{p^2}$, to find the endomorphism ring $\mathrm{End}(E)$. Wesolowski [43] showed that this is equivalent to the isogeny problem under reductions of polynomial expected time, assuming the generalised Riemann hypothesis, and further, Page and Wesolowski [33] recently showed that the endomorphism ring problem is equivalent to the problem of computing one endomorphism.

2.2 Quaternion algebras and the Deuring correspondence

We give the necessary arithmetic background to understand the signing procedure of SQIsign at a high level.[5] The heart of the signing procedure in SQIsign lies in the Deuring correspondence, which connects the geometric world of supersingular curves from Section 2.1 to the arithmetic world of quaternion algebras. For more details on quaternion algebras, we refer to Voight [42].

Quaternion algebras, orders and ideals. Quaternion algebras are four-dimensional $\mathbb{Q}$-algebras, generated by four elements $1, i, j, k$ following certain multiplication rules. For SQIsign, we work in the quaternion algebra *ramified* at p and ∞. When $p \equiv 3 \pmod 4$, one representation of such a quaternion algebra is given by $\mathcal{B}_{p,\infty} = \mathbb{Q} + i\mathbb{Q} + j\mathbb{Q} + k\mathbb{Q}$ with multiplication rules

$$i^2 = -1,\ j^2 = -p,\ ij = -ji = k.$$

For an element $\alpha = x + yi + zj + wk \in \mathcal{B}_{p,\infty}$ with $x, y, z, w \in \mathbb{Q}$, we define its *conjugate* to be $\bar{\alpha} = x - yi - zj - wk$, and its *reduced norm* to be $n(\alpha) = \alpha\bar{\alpha}$.

We are mainly interested in *lattices* in $\mathcal{B}_{p,\infty}$, defined as full-rank $\mathbb{Z}$-modules contained in $\mathcal{B}_{p,\infty}$, i.e., abelian groups of the form

$$\alpha_1\mathbb{Z} + \alpha_2\mathbb{Z} + \alpha_3\mathbb{Z} + \alpha_4\mathbb{Z},$$

where $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \in \mathcal{B}_{p,\infty}$ are linearly independent. If a lattice $\mathcal{O} \subset \mathcal{B}_{p,\infty}$ is also a subring of $\mathcal{B}_{p,\infty}$, i.e., it contains 1 and is closed under multiplication, then $\mathcal{O}$ is called an *order*. Orders that are not strictly contained in any other order are called *maximal* orders. From this point on, we only consider maximal orders.

A lattice I that is closed under multiplication by an order $\mathcal{O}$ on the left is called a *left* (resp. *right*) *$\mathcal{O}$-ideal*. We refer to $\mathcal{O}$ as the left (resp. right) order of I. When $\mathcal{O}$ is the left order of I and $\mathcal{O}'$ the right order of I, we define I to be a *connecting* $(\mathcal{O}, \mathcal{O}')$*-ideal*.[6] A left $\mathcal{O}$-ideal I that is also contained in $\mathcal{O}$ is called an *integral* ideal. From this point on, we only deal with integral left ideals and simply refer to them as ideals.

The *norm* of an ideal I is the greatest common divisor of the reduced norms of the elements of I, whereas the *conjugate* $\bar{I}$ of an ideal I is the ideal consisting of the conjugates

[5] This section is only necessary for Section 2.3 and Section 3, as all other sections are concerned only with SQIsign verification, which will only use well-known isogeny terminology. In contrast, signing heavily relies on the arithmetic of quaternion algebras.

[6] Note that $\mathcal{O}$ and $\mathcal{O}'$ need not be distinct.

of the elements of I. Two ideals I and J are said to be equivalent if $I = J\alpha$ for some $\alpha \in \mathcal{B}_{p,\infty}^{\times}$ and is denoted $I \sim J$. Equivalent ideals have equal left orders and isomorphic right orders.

The Deuring correspondence. Given an elliptic curve E with $\mathrm{End}(E) \cong \mathcal{O}$, there is a one-to-one correspondence between separable isogenies from E and left $\mathcal{O}$-ideals I of norm coprime to p. Given an isogeny φ, we denote the corresponding ideal I_φ, and conversely, given an ideal I, we denote the corresponding isogeny φ_I. The Deuring correspondence acts like a dictionary: a given isogeny $\varphi : E \to E'$ corresponds to an ideal I_φ with left order $\mathcal{O} \cong \mathrm{End}(E)$ and right order $\mathcal{O}' \cong \mathrm{End}(E')$. Furthermore, the degree of φ is equal to the norm of I_φ and the dual isogeny $\widehat{\varphi}$ corresponds to the conjugate $\overline{I_\varphi} = I_{\widehat{\varphi}}$. Equivalent ideals I, J have isomorphic right orders and the corresponding isogenies φ_I, φ_J have isomorphic codomains.

The (generalised) KLPT algorithm. The KLPT algorithm, introduced by Kohel, Lauter, Petit, and Tignol [30], is a purely quaternionic algorithm, but has seen a variety of applications in isogeny-based cryptography due to the Deuring correspondence. Given an ideal I, KLPT finds an equivalent ideal J of prescribed norm. The drawback is that the norm of the output J will be comparatively large.

For example, the KLPT algorithm is used to compute isogenies between two curves of known endomorphism ring. Given two maximal orders $\mathcal{O}, \mathcal{O}'$, translating the standard choice[7] of connecting ideal I to its corresponding isogeny is hard. However, by processing I through KLPT first, we can find an equivalent ideal J of smooth norm, allowing us to compute φ_J. This is essential for computing the response in SQIsign.

The original KLPT algorithm only works for $\mathcal{O}_0$-ideals, where $\mathcal{O}_0$ is a maximal order of a special form.[8] This was generalised by De Feo, Kohel, Leroux, Petit, and Wesolowski [21] to work for arbitrary orders $\mathcal{O}$, albeit at the cost of an even larger norm bound for the output. Note that SQIsign utilizes both versions.

2.3 SQIsign

Next, we give a high-level description of signing and verification in SQIsign. SQIsign is a signature scheme based on an underlying Sigma protocol that proves knowledge of a *secret* (non-scalar) endomorphism $\alpha \in \mathrm{End}(E_A)$ for some *public* curve E_A. At its core, the prover shows this knowledge by being able to compute an isogeny φ from E_A to some random curve E_2.

A high-level description of the SQIsign Sigma protocol is given below(see also Figure 1)
.

[7] $I = N\mathcal{O}\mathcal{O}'$, where N is the smallest integer making I integral.

[8] Specifically, it only works for special, p-extremal orders. An example of such an order when $p \equiv 3 \pmod 4$ is $\mathrm{End}(E_0)$ where $j(E_0) = 1728$.

$$\begin{tikzcd}
 & E_0 \arrow[r, "\varphi_{\text{com}}"'] \arrow[dl, dashed, "\varphi_A"] & E_1 \arrow[d, "\varphi_{\text{chall}}"'] \\
E_A \arrow[rr, "\varphi_{\text{resp}}"] & & E_2
\end{tikzcd}$$

Fig. 1: The SQIsign protocol with three phases: commitment φ_{com}, challenge φ_{chall}, and response φ_{resp}.

Setup: Fix a prime number p and supersingular elliptic curve $E_0/\mathbb{F}_{p^2}$ with known endomorphism ring.

Key generation: Compute a secret key $\varphi_A : E_0 \to E_A$, giving the prover knowledge of $\text{End}(E_A)$, with corresponding public verification key E_A.

Commit: The prover generates a random commitment isogeny $\varphi_{\text{com}} : E_0 \to E_1$, and sends E_1 to the verifier.

Challenge: The verifier computes a random challenge isogeny $\varphi_{\text{chall}} : E_1 \to E_2$, and sends φ_{chall} to the prover.

Response: The prover uses the knowledge of φ_{com} and φ_{chall} to compute $\text{End}(E_2)$, allowing the prover to compute the response isogeny $\varphi_{\text{resp}} : E_A \to E_2$, by translating an ideal computed using the generalised KLPT algorithm, as described at the end of Section 2.2.

The verifier needs to check that φ_{resp} is an isogeny from E_A to E_2.[9] Assuming the hardness of the endomorphism ring problem, the protocol is sound: if the prover is able to respond to two different challenges φ_{chall}, φ'_{chall} with φ_{resp} and φ'_{resp}, the prover knows an endomorphism of the public key E_A, namely $\widehat{\varphi'_{\text{resp}}} \circ \varphi'_{\text{chall}} \circ \widehat{\varphi_{\text{chall}}} \circ \varphi_{\text{resp}}$. Proving zero-knowledge is harder and relies on the output distribution of the generalised KLPT algorithm. Note that KLPT is needed for computing the response:[10] while setting $\varphi_{\text{resp}} = \varphi_{\text{chall}} \circ \varphi_{\text{com}} \circ \widehat{\varphi_A}$ gives an isogeny from E_A to E_2, this leaks the secret φ_A.[11] For a further discussion on zero-knowledge, we refer to the original SQIsign articles [21, 22].

Remark 1. The best-known attacks against SQIsign are the generic attacks against the endomorphism ring problem. As discussed in Section 2.1, their run time depends only on the size of p and, with high probability, do not recover the original secret isogeny, but

[9] Additionally, $\widehat{\varphi_{\text{chall}}} \circ \varphi_{\text{resp}}$ needs to be cyclic. Observe that otherwise, the soundness proof might return a scalar endomorphism.

[10] Alternatively, one can replace the connecting ideal with the shortest equivalent ideal, and translate it by embedding it in an isogeny between higher-dimensional abelian varieties, as shown in SQIsignHD [19]

[11] Further, this is not a valid response, since the composition with $\widehat{\varphi_{\text{chall}}}$ is not cyclic.

rather a different isogeny between the same curves. Therefore, their complexity should be unaffected by the changes we introduce to the SQIsign protocol in Section 3, as for these attacks it does not matter whether the original secret isogeny had kernel points defined over a larger extension field. In short, the changes in this work do not affect the security of SQIsign.

Verification. Using the Fiat–Shamir heuristic, the SQIsign Sigma protocol is transformed into a signature scheme. This means that a signature on the message msg is of the form $\sigma = (\varphi_{\mathrm{resp}}, \mathtt{msg}, E_1)$. For efficiency, φ_{resp} is compressed, and E_1 is replaced by a description of φ_{chall}. Thus, given the signature σ and public key E_A, the verifier recomputes the response isogeny $\varphi_{\mathrm{resp}} : E_A \to E_2$ and the (dual of the) challenge isogeny $\widehat{\varphi_{\mathrm{chall}}} : E_2 \to E_1$, and then verifies that the hash $H(\mathtt{msg}, E_1)$ indeed generates φ_{chall}.

The isogeny φ_{resp} is of degree 2^e with $e = \lceil \frac{15}{4} \log(p) \rceil + 25$ [12, §7.2.3], where 2^e corresponds to the output size of the generalised KLPT algorithm. The bottleneck in verification is the (re)computation of φ_{resp} in $\lceil e/f \rceil$ steps of size 2^f. Accelerating this will be the focus of this paper.

2.4 SQIsign-friendly primes

Next, we give more details on the parameter requirements in SQIsign. We refer to the original SQIsign works [12, 21, 22] for a detailed description of their origins.

SQIsign prime requirements. The main bottleneck of SQIsign is the computation of isogenies. Recall from Equations (1) and (2) that, when working with supersingular elliptic curves $E/\mathbb{F}_{p^2}$, we have $E(\mathbb{F}_{p^2}) = E[p \pm 1]$. Thus, to use x-only arithmetic over $\mathbb{F}_{p^2}$, SQIsign restricts to computing isogenies of *smooth* degree $N \mid (p^2 - 1)$. Finding SQIsign-friendly primes reduces to finding primes p, with $p^2 - 1$ divisible by a large, smooth number. More explicitly, for a security level λ, the following parameters are needed:

- A prime p of bitsize $\log_2(p) \approx 2\lambda$ with $p \equiv 3 \mod 4$.
- The torsion group $E[2^f]$ as large as possible, that is $2^f \mid p + 1$.
- A smooth odd factor $T \mid (p^2 - 1)$ of size roughly $p^{5/4}$.
- The degree of φ_{com}, $D_{\mathrm{com}} \mid T$, of size roughly $2^{2\lambda} \approx p$.
- The degree of φ_{chall}, $D_{\mathrm{chall}} \mid 2^f T$, of size roughly $2^{\lambda} \approx p^{1/2}$.
- Coprimality between D_{com} and D_{chall}.

To achieve NIST Level I, III, and V security, we set the security parameter as $\lambda = 128, 192, 256$, respectively. Concretely, this means that, for each of these security parameters, we have $\log p \approx 256, 384, 512$, and $\log T \approx 320, 480, 640$, with f as large as possible given the above restrictions. The smoothness of T directly impacts the signing time, and the problem of finding primes p with a large enough T that is reasonably smooth is difficult. We refer to recent work on this problem for techniques to find suitable primes [10, 12, 14, 18, 21, 22].

The crucial observation for this work is that T occupies space in $p^2 - 1$ that limits the size of f, hence current SQIsign primes balance the smoothness of T with the size of f.

Remark 2. SQIsign (NIST) further requires $3^g \mid p+1$ such that $D_{\text{chall}} = 2^f \cdot 3^g \geq p^{1/2}$ and $D_{\text{chall}} \mid p+1$. While this is not a strict requirement in the theoretical sense, it facilitates efficiency of computing φ_{chall}. From this point on, we ensure that this requirement is always fulfilled.

Remark 3. Since the curves E and their twists E^t satisfy

$$E(\mathbb{F}_{p^2}) \cong \mathbb{Z}/(p \pm 1)\mathbb{Z} \oplus \mathbb{Z}/(p \pm 1)\mathbb{Z},$$

and we work with both simultaneously, choosing T and f is often incorrectly described as choosing divisors of $p^2 - 1$. There is a subtle issue here: even if 2^f divides $p^2 - 1$, $E[2^f]$ may not exist as a subgroup of $\langle E(\mathbb{F}_{p^2}), \rho^{-1}(E^t(\mathbb{F}_{p^2}))\rangle \subseteq E(\mathbb{F}_{p^4})$, where $\rho : E \to E^t$ is the twisting isomorphism. While this does not usually matter in the case of SQIsign (we pick 2^f as a divisor of $p+1$, and T is odd), this becomes a problem when working over multiple extension fields. In Section 3.2, we make this precise and reconcile it using Theorem 1.

2.5 Computing rational isogenies from irrational generators

Finally, to facilitate signing with field extensions, we recall the techniques for computing $\mathbb{F}_{p^2}$-rational isogenies, i.e., isogenies defined over $\mathbb{F}_{p^2}$, generated by *irrational* kernel points, that is, not defined over $\mathbb{F}_{p^2}$. In the context of this paper, we again stress that such isogenies will only be computed by the signer; the verifier will only work with points in $\mathbb{F}_{p^2}$.

The main computational task of most isogeny-based cryptosystems (including SQIsign) lies in evaluating isogenies given the generators of their kernels. Explicitly, given an elliptic curve $E/\mathbb{F}_q$, a point $K \in E(\mathbb{F}_{q^k})$ such that $\langle K\rangle$ is *defined over* $\mathbb{F}_q$,[12] and a list of points $(P_1, P_2, \ldots, P_n)$ in E, we wish to compute the list of points $(\varphi(P_1), \varphi(P_2), \ldots, \varphi(P_n))$, where φ is the separable isogeny with $\ker \varphi = \langle K\rangle$. Since we work with curves E whose p^2-Frobenius π is equal to the multiplication-by-$(-p)$ map (see Section 2.1), *every* subgroup of E is closed under the action of $\mathrm{Gal}(\bar{\mathbb{F}}_{p^2}/\mathbb{F}_{p^2})$, hence every isogeny from E can be made $\mathbb{F}_{p^2}$-rational, by composing with the appropriate isomorphism.

Computing isogenies of smooth degree. Recall from Section 2.1 that the isogeny factors as a composition of small prime degree isogenies, which we compute using Vélu-style algorithms. For simplicity, for the rest of the section, we therefore assume that $\langle K\rangle$ is a subgroup of order $\ell > 2$, where ℓ is a small prime.

[12] That is, the group $\langle K\rangle$ is closed under the action of $\mathrm{Gal}(\bar{\mathbb{F}}_q/\mathbb{F}_q)$.

At the heart of these Vélu-style isogeny formulas is evaluating the kernel polynomial. Pick any subset $S \subseteq \langle K \rangle$ such that $\langle K \rangle = S \sqcup -S \sqcup \{\infty\}$. Then the kernel polynomial can be written as

$$f_S(X) = \prod_{P \in S} (X - x(P)). \tag{3}$$

Here, the generator K can be either a rational point, i.e., lying in $E(\mathbb{F}_q)$, or an irrational point, i.e., lying in $E(\mathbb{F}_{q^k})$ for $k > 1$, but whose group $\langle K \rangle$ is defined over $\mathbb{F}_q$. Next, we discuss how to solve the problem efficiently in the latter case.

Irrational generators. For $K \notin E(\mathbb{F}_q)$ of order ℓ, we can speed up the computation of the kernel polynomial using the action of Frobenius. This was used in two recent works [6, 24], though the general idea was used even earlier [40].

As $\langle K \rangle$ is defined over $\mathbb{F}_q$, we know that the q-power Frobenius π acts as an endomorphism on $\langle K \rangle \subseteq E(\mathbb{F}_{p^k})$ and thus maps K to a multiple $[\gamma]K$ for some $\gamma \in \mathbb{Z}$. This fully determines the action on $\langle K \rangle$, i.e., $\pi|_{\langle K \rangle}$ acts as $P \mapsto [\gamma]P$ for all $P \in \langle K \rangle$. For the set S as chosen above, this action descends to an action on its x-coordinates $X_S = \{x(P) \in \mathbb{F}_{q^k} \mid P \in S\}$ and thus partitions X_S into orbits $\{x(P), x([\gamma]P), x([\gamma^2]P), \ldots\}$ of size equal to the order of γ in $(\mathbb{Z}/\ell\mathbb{Z})^\times/\{1, -1\}$.

If we pick one representative $P \in S$ per orbit, and call this set of points S_0, we can compute the kernel polynomial (3) as a product of the minimal polynomials $\mu_{x(P)}$ of the $x(P) \in \mathbb{F}_{q^k}$ for these $P \in S_0$, with each $\mu_{x(P)}$ defined over $\mathbb{F}_q$, as

$$f_S(X) = \prod_{P \in S_0} \mu_{x(P)}(X), \tag{4}$$

where μ_β denotes the minimal polynomial of β over $\mathbb{F}_q$.

To compute $f_S(\alpha)$ for $\alpha \in \mathbb{F}_q$, we only require the smaller polynomial $f_{S_0}(X)$ and compute $\mathrm{Norm}_{\mathbb{F}_{q^k}/\mathbb{F}_q}(f_{S_0}(\alpha))$, as

$$\mathrm{Norm}_{\mathbb{F}_{q^k}/\mathbb{F}_q}(f_{S_0}(\alpha)) = \prod_{\pi \in G} \pi(f_{S_0}(\alpha)) = \prod_{P \in S_0} \prod_{\pi \in G} (\alpha - \pi(x(P))) = \prod_{P \in S_0} \mu_{x(P)}(\alpha),$$

where $G = \mathrm{Gal}(\mathbb{F}_{q^k}/\mathbb{F}_q)$, as per Banegas, Gilchrist, Le Dévéhat, and Smith [6]. This allows us to compute the image under f_S of x-values of points in $E(\mathbb{F}_q)$, but only works for values in $\mathbb{F}_q$. To evaluate $f_S(\alpha)$ for general $\alpha \in \overline{\mathbb{F}}_p$, i.e. to compute the image of a point in $E(\overline{\mathbb{F}}_p)$, we instead use the larger polynomial $f_S(X)$, which we compute, as in Equation (4), as a product of the minimal polynomials $\mu_{x(P)}$, where we use Shoup's algorithm [37] to compute each $\mu_{x(P)}$ given $x(P)$. Computing $f_S(X)$ requires a total of $O(\ell k) + \widetilde{O}(\ell)$ operations, with k such that each $x(P) \in \mathbb{F}_{q^k}$. Evaluation f_S at α takes $\widetilde{O}(\ell k')$ operations, with k' the smallest value such that $\alpha \in \mathbb{F}_{q^{k'}}$ [24, Section 4.3].

Remark 4. The biggest drawback to using this technique is that $\sqrt{\text{élu}}$ is no longer effective, as we would need to work in the smallest field where both the isogeny generator and the x-value of the point we are evaluating are defined in.

3 Signing with extension fields

By allowing torsion T from extension fields, we enable more flexibility in choosing SQIsign primes p, thus enabling a larger $2^{\bullet}$-torsion. Such torsion T requires us to compute rational isogenies with kernel points in extension fields $\mathbb{F}_{p^{2k}}$. This section describes how to adapt SQIsign's signing procedure to enable such isogenies, and the increased cost this incurs. In particular, we describe two approaches for T: allowing torsion T from a particular extension field $\mathbb{F}_{p^{2k}}$, or from all extension fields $\mathbb{F}_{p^{2n}}$ for $1 \leq n \leq k$. The first approach means that we can look for T dividing an integer of bit size $\Theta(k \log p)$, and the second approach allows for $\Theta(k^2 \log p)$. In Section 4, we explore how increased $2^{\bullet}$-torsion affects verification.

Throughout this section, we will reuse the notation from Section 2.4 to describe the various parameters related to SQIsign.

3.1 Changes in the signing procedure

Recall from Section 2.3 that the signing operation in SQIsign requires us to work with both elliptic curves and quaternion algebras, and to translate back and forth between these worlds. Note that the subroutines that work solely with objects in the quaternion algebra $\mathcal{B}_{p,\infty}$, including all operations in KLPT and its derivatives, are indifferent to what extension fields the relevant torsion groups lie in. Hence, a large part of signing is unaffected by torsion from extension fields.

In fact, the only subroutines that are affected by moving to extension fields are those relying on Algorithm 1, $\mathsf{IdealToIsogeny}_D$, which translates $\mathcal{O}_0$-ideals I of norm dividing D to their corresponding isogenies φ_I. $\mathsf{IdealToIsogeny}_D$ is not used during verification, and is used only in the following parts of signing:

Commitment: The signer translates a random ideal of norm D_{com} to its corresponding isogeny, using one execution of $\mathsf{IdealToIsogeny}_{D_{\text{com}}}$.

Response: The signer translates an ideal of norm 2^e to its corresponding isogeny, requiring $2 \cdot \lceil e/f \rceil$ executions of $\mathsf{IdealToIsogeny}_T$.[13]

Remark 5. We will choose parameters such that $2^f \mid p+1$ and $D_{\text{chall}} \mid p+1$, so that $E[2^f]$ and $E[D_{\text{chall}}]$ are defined over $\mathbb{F}_{p^2}$. As a result, the verifier only works in $\mathbb{F}_{p^2}$ and the added complexity of extension fields applies only to the signer.

[13] The technical details are given by De Feo, Leroux, Longa, and Wesolowski [22].

Algorithm 1 IdealToIsogeny$_D(I)$

```
Input: $I$ a left $\mathcal{O}_0$-ideal of norm dividing $D$
Output: $\varphi_I$
 1: Compute $\alpha$ such that $I = \mathcal{O}_0\langle\alpha, \mathrm{nrd}(I)\rangle$
 2: Let $\mathbf{A} = [1, i, \frac{i+j}{2}, \frac{1+k}{2}]$ denote a basis of $\mathcal{O}_0$
 3: Compute $\mathbf{v}_{\bar{\alpha}} := [x_1, x_2, x_3, x_4]^T \in \mathbb{Z}^4$ such that $\mathbf{A}\mathbf{v}_{\bar{\alpha}} = \bar{\alpha}$
 4: for $\ell^e \mid\mid D$ do
 5:     $\bar{\alpha}|_{\langle P_{\ell^e}, Q_{\ell^e}\rangle} := x_1\mathbf{I} + x_2(i|_{\langle P_{\ell^e}, Q_{\ell^e}\rangle}) + x_3(\frac{i+j}{2}|_{\langle P_{\ell^e}, Q_{\ell^e}\rangle}) + x_4(\frac{1+k}{2}|_{\langle P_{\ell^e}, Q_{\ell^e}\rangle})$
 6:     Let $a, b, c, d$ be integers such that $\bar{\alpha}|_{\langle P_{\ell^e}, Q_{\ell^e}\rangle} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$
 7:     $K_{\ell^e} := [a]P_{\ell^e} + [c]Q_{\ell^e}$
 8:     if $\mathrm{ord}(K_{\ell^e}) < \ell^e$ then
 9:         $K_{\ell^e} = [b]P_{\ell^e} + [d]Q_{\ell^e}$
10: Set $\varphi_I$ to be the isogeny generated by the points $K_{\ell^e}$.
11: return $\varphi_I$
```

Adapting ideal-to-isogeny translations to field extensions. To facilitate signing with field extensions, we slightly adapt IdealToIsogeny$_D$ so that it works with prime powers separately. Note that the additional cost of this is negligible compared to the cost of computing the isogeny from the generators because finding the action of the relevant endomorphisms consists of simple linear algebra. See Algorithm 1 for details.

In Line 5 of Algorithm 1, the notation $\beta|_{\langle P_{\ell^e}, Q_{\ell^e}\rangle}$ refers to the action of an endomorphism β on a fixed basis P_{ℓ^e}, Q_{ℓ^e} of $E[\ell^e]$. This action is described by a matrix in $M_2(\mathbb{Z}/\ell^e\mathbb{Z})$. These matrices can be precomputed, hence the only operations in which the field of definition of $E[\ell^e]$ matters are the point additions in Lines 7 and 9, and isogenies generated by each K_{ℓ^e} in Line 10.

3.2 Increased torsion availability from extension fields

Next, we detail the two approaches to allow torsion groups from extension fields, which permits more flexibility in choosing the final prime p.

Working with a single field extension of $\mathbb{F}_{p^2}$. Although the choice of solely working in $\mathbb{F}_{p^2}$ occurs naturally,[14] there is no reason *a priori* that this choice is optimal. Instead, we can choose to work in the field $\mathbb{F}_{p^{2k}}$. We emphasise that this does *not* affect signature sizes; the only drawback is that we now perform more expensive $\mathbb{F}_{p^{2k}}$-operations during signing in IdealToIsogeny. The upside, however, is a relaxed prime requirement: we are no

[14] It is the smallest field over which every isomorphism class of supersingular elliptic curves has a model.

longer bound to $E[T] \subseteq \langle E(\mathbb{F}_{p^2}), \rho^{-1}(E^t(\mathbb{F}_{p^2}))\rangle$ and can instead use

$$E[T] \subseteq \langle E(\mathbb{F}_{p^{2k}}), \rho^{-1}(E^t(\mathbb{F}_{p^{2k}}))\rangle.$$

By Equations (1) and (2), we have $E(\mathbb{F}_{p^{2k}}) \cong E[p^k \pm 1]$ and $E^t(\mathbb{F}_{p^{2k}}) \cong E[p^k \mp 1]$, thus we simply get

$$E[T] \subseteq E\left[\frac{p^{2k}-1}{2}\right],$$

since $\langle E[A], E[B]\rangle = E[\mathrm{lcm}(A,B)]$. Hence, by using torsion from $E(\mathbb{F}_{p^{2k}})$, we increase $T \mid (p^2-1)/2$ to $T \mid (p^{2k}-1)/2$. This implies there are $2(k-1)\log p$ more bits available to find T with adequate smoothness.

Working with multiple field extensions of $\mathbb{F}_{p^2}$. Instead of fixing a single higher extension field $\mathbb{F}_{p^{2k}}$, we can also choose to work with multiple field extensions, in particular all fields $\mathbb{F}_{p^{2n}}$, where $1 \leq n \leq k$. The torsion group we can access by this relaxed requirement is described by the following definition.

Definition 1. *Let E be a supersingular elliptic curve over $\mathbb{F}_{p^2}$ and let E^t_n denote an arbitrary quadratic twist of E over $\mathbb{F}_{p^{2n}}$ with respect to the twisting isomorphism $\rho_n : E \to E^t_n$. We define the k-*available torsion *of E to be the group generated by $E(\mathbb{F}_{p^{2n}})$ and $\rho_n^{-1}(E^t_n(\mathbb{F}_{p^{2n}}))$ for $1 \leq n \leq k$.*

Any point P in the k-available torsion can thus be written as a sum

$$P = \sum_{i=1}^{k} (P_i + \rho_n^{-1}(P_i^t))$$

of points $P_i \in E(\mathbb{F}_{p^{2n}})$ and $P_i^t \in E^t_n(\mathbb{F}_{p^{2n}})$. Since the twisting isomorphism keeps the x-coodinate fixed, the computation of this isomorphism can be ignored when using x-only arithmetic, and we simply obtain a sum of points whose x-coordinates lie in $\mathbb{F}_{p^{2n}}$ for $1 \leq n \leq k$. This justifies the name k-available torsion, as we do not have to go beyond $\mathbb{F}_{p^{2k}}$ to do arithmetic with P by working with the summands separately.

The structure of the k-available torsion is completely captured by the following result.

Theorem 1. *Let $p > 2$ be a prime, and let $E/\mathbb{F}_{p^2}$ be a supersingular curve with $\mathrm{tr}(\pi) = \pm 2p$, where π is the Frobenius endomorphism. Then the k-available torsion is precisely the group $E[N]$ with*

$$N = \prod_{n=1}^{k} \Phi_n(p^2)/2,$$

where Φ_n denotes the n-th cyclotomic polynomial.

Lemma 1. *For any integer $m \geq 2$, we have the following identity*

$$\operatorname{lcm}\left(\{m^n - 1\}_{n=1}^{k}\right) = \prod_{n=1}^{k} \Phi_n(m)$$

where Φ_n denotes the n-th cyclotomic polynomial.

Proof. We denote the left-hand side and right-hand side of the equation in the statement of the lemma by LHS and RHS, respectively. We show that any prime power dividing one side, also divides the other.

For any prime ℓ and $e > 0$, if ℓ^e divides the LHS, then, by definition, it divides $m^i - 1 = \prod_{d|i} \Phi_d(m)$ for some $1 \leq i \leq k$. Hence, it also divides the RHS. Conversely, if ℓ^e divides the RHS, then ℓ^e also divides the LHS. To show this, we need to know when $\Phi_i(m)$ and $\Phi_j(m)$ are coprime. We note that

$$\gcd(\Phi_i(m), \Phi_j(m)) \mid R$$

where R is the resultant of $\Phi_i(X)$ and $\Phi_j(X)$, and a classic result by Apostol [2, Theorem 4.], tells us that

$$\operatorname{Res}(\Phi_i(X), \Phi_j(X)) > 1 \Rightarrow i = jm$$

for $i > j$ and some integer m.

Using this, if ℓ^e divides the RHS, then it will also divide the product

$$\prod_{n=1}^{\lfloor k/d \rfloor} \Phi_{dn}(m),$$

for some integer d, and this product divides the LHS, as it divides $m^{d\lfloor k/d \rfloor} - 1$. □

We can now conclude the proof of Theorem 1.

Proof. From the structure of $E(\mathbb{F}_{p^{2n}})$ (see Remark 3), where E is as in the statement, the k-available torsion can be seen as the group generated by the full torsion groups

$$E[p^n \pm 1]$$

for $1 \leq n \leq k$. Using the fact that

$$\langle E[A], E[B] \rangle = E[\operatorname{lcm}(A, B)],$$

we see that the k-available torsion is $E[N]$ where

$$N = \operatorname{lcm}\left(\{p^n - 1\}_{n=1}^{k} \cup \{p^n + 1\}_{n=1}^{k}\right) = \operatorname{lcm}\left(\{p^{2n} - 1\}_{n=1}^{k}\right)/2,$$

where the last equality only holds for $p > 2$. Applying Lemma 1 with $m = p^2$, we obtain

$$N = \prod_{k=1}^{n} \Phi_k(p^2)/2,$$

□

We find that using all extension fields $\mathbb{F}_{p^{2n}}$, for $1 \leq n \leq k$, increases $T \mid p^2 - 1$ to $T \mid N$, with N as given by Theorem 1. Given that

$$\log(N) = \sum_{n=1}^{k} \log(\Phi_n(p^2)/2) \approx 2 \sum_{n=1}^{k} \phi(n) \log(p),$$

and the fact that $\sum_{n=1}^{k} \phi(n)$ is in the order of $\Theta(k^2)$, where ϕ denotes Euler's totient function, we find that $T \mid N$ gives roughly $k^2 \log(p)$ more bits to find T with adequate smoothness, compared to the $\log(p)$ bits in the classical case of working over $\mathbb{F}_{p^2}$, and $k \log(p)$ bits in the case of working over $\mathbb{F}_{p^{2k}}$. Due to this, we only consider working in multiple field extensions from this point on.

3.3 Cost of signing using extension fields

In SQIsign, operations over $\mathbb{F}_{p^2}$ make up the majority of the cost during signing [22, Section 5.1]. Hence, we can roughly estimate the cost of signing by ignoring purely quaternionic operations, in which case the bottleneck of the signing procedure becomes running $\mathsf{IdealToIsogeny}_T$ as many times as required by the IdealToIsogenyEichler algorithm [22, Algorithm 5] in the response phase. In other words, we estimate the total signing cost from the following parameters:

- f, such that $2^f \mid p+1$.
- T, the chosen torsion to work with.
- For each $\ell_i^{e_i} \mid T$, the smallest k_i such that $E[\ell_i^{e_i}]$ is defined over $\mathbb{F}_{p^{2k_i}}$.

Since Algorithm 1 works with prime powers separately, we can estimate the cost of a single execution by considering the cost per prime power.

Cost per prime power. For each $\ell_i^{e_i} \mid T$, let k_i denote the smallest integer so that $E[\ell_i^{e_i}] \subseteq E(\mathbb{F}_{p^{2k_i}})$, and let $M(k_i)$ denote the cost of operations in $\mathbb{F}_{p^{2k_i}}$ in terms of $\mathbb{F}_{p^2}$-operations. Computing the generator $K_{\ell_i^{e_i}}$ consists of a few point additions in $E[\ell_i^{e_i}]$, hence is $O(M(k) \cdot e \log \ell)$, while the cost of computing the isogeny generated by $K_{\ell_i^{e_i}}$ comes from computing e isogenies of degree ℓ at a cost of $O(\ell k) + \tilde{O}(\ell)$, using the techniques from Section 2.5.

To compute the whole isogeny, we need to push the remaining generators $K_{\ell_j^{e_j}}$, through this isogeny. To minimize the total cost, we pick the greedy strategy of always computing the smaller ℓ first. This bounds the cost of evaluating K_{ℓ^e} in *other* isogenies by $O(M(k) \cdot \ell)$.

Total cost of signing. Based on the analysis above, we let

$$\textsc{Cost}_p(\ell_i^{e_i}) = c_1 M(k_i) e \log \ell + c_2 e_i \ell_i k_i + c_3 e_i \ell_i \log(\ell_i) + c_4 M(k_i) \ell$$

where k_i, and $M(k)$ are as before, and c_i are constants corresponding to the differences in the asymptotic complexities. Since we can estimate the total cost of executing $\mathsf{IdealToIsogeny}_T$ by summing the cost of each maximal prime power divisor of T, and observing that signing consists of executing $\mathsf{IdealToIsogeny}_{D_{\mathsf{com}}}$ one time, and $\mathsf{IdealToIsogeny}_T$ a total of $2 \cdot \lceil e/f \rceil$ times, we get a rough cost estimate of signing as

$$\textsc{SigningCost}_p(T) = (2 \cdot \lceil e/f \rceil + 1) \cdot \sum_{\ell_i^{e_i} | T} \textsc{Cost}_p(\ell_i^{e_i}).$$

In Section 7, we use this function to pick p and T minimising this cost. While this cost metric is very rough, we show that our implementation roughly matches the times predicted by this function. Further, we show that this cost metric suggests that going to extension fields gives signing times within the same order of magnitude as staying over $\mathbb{F}_{p^2}$, even when considering the additional benefit of using $\sqrt{\text{élu}}$ to compute isogenies in the latter case.

4 Effect of increased $2^\bullet$-torsion on verification

In Section 3, we showed that signing with extension fields gives us more flexibility in choosing the prime p, and, in particular, allows us to find primes with rational 2^f-torsion for larger f. In this section, we analyse how such an increase in $2^\bullet$-torsion affects the cost of SQIsign verification, e.g., computing φ_{resp} and $\widehat{\varphi_{\text{chall}}}$, in terms of $\mathbb{F}_p$-multiplications,[15] taking the SQIsign (NIST) implementation (with no further optimisations) as the baseline for comparison.

4.1 Detailed description of verification

Before giving an in-depth analysis of verification performance, we give a detailed description of how verification is executed. Recall that a SQIsign signature σ for a message `msg` created by a signer with secret signing key $\varphi_A : E_0 \to E_A$ proves knowledge of an endomorphism on E_A by describing an isogeny $\varphi_{\text{resp}} : E_A \to E_2$ of degree 2^e(see Figure 1). A given message `msg` is hashed on E_1 to a point K_{chall} of order D_{chall}, hence represents an isogeny $\varphi_{\text{chall}} : E_1 \to E_2$. A signature is valid if the composition of φ_{resp} with $\widehat{\varphi_{\text{chall}}}$ is cyclic of degree $2^e \cdot D_{\text{chall}}$.

Thus, to verify a signature σ, the verifier must **(a)** recompute φ_{resp}, **(b)** compute the dual of φ_{chall}, to confirm that both are well-formed, and finally **(c)** recompute the hash of the message `msg` to confirm the validity of the signature.

[15] As standard, we denote multiplications by **M**, squarings by **S**, and additions by **a**.

In SQIsign, the size of the sample space for φ_{chall} impacts soundness, a key security property for signature schemes. In SQIsign (NIST), to obtain negligible soundness error (in the security parameter λ) the message is hashed to an isogeny of degree $D_{\text{chall}} = 2^f \cdot 3^g$ so that the size of cyclic isogenies of degree D_{chall} is larger than 2^λ. In contrast, when $f \geq \lambda$, we can simply set $D_{\text{chall}} = 2^\lambda$.

The signature σ consists of a compressed description of the isogenies φ_{resp} and $\widehat{\varphi_{\text{chall}}}$. For $f < \lambda$ and $D_{\text{chall}} = 2^f \cdot 3^g$ it is of the form

$$\sigma = (b, s^{(1)}, \dots, s^{(n)}, r, b_2, s_2, b_3, s_3)$$

with $s^{(j)}, s_2 \in \mathbb{Z}/2^f\mathbb{Z}$, $s_3 \in \mathbb{Z}/3^g\mathbb{Z}$, $r \in \mathbb{Z}/2^f 3^g\mathbb{Z}$, and $b, b_2, b_3 \in \{0, 1\}$. If $f \geq \lambda$, we set $D_{\text{chall}} = 2^f$ and have $s_2 \in \mathbb{Z}/2^\lambda\mathbb{Z}$ and $r \in \mathbb{Z}/2^f\mathbb{Z}$, while b_3, s_3 are omitted.
Algorithmically, the verification process mostly requires three subroutines.

FindBasis: Given a curve E, find a deterministic basis (P, Q) of $E[2^f]$.

FindKernel: Given a curve E with basis (P, Q) for $E[2^f]$ and $s \in \mathbb{Z}/2^f\mathbb{Z}$, compute the kernel generator $K = P + [s]Q$.

ComputeIsogeny: Given a curve E and a kernel generator K, compute the isogeny $\varphi : E \to E/\langle K \rangle$ and $\varphi(Q)$ for some $Q \in E$.

Below we detail each of the three verification steps **(a)**-**(c)**.

Step (a). Computing φ_{resp} is split up into $n - 1$ *blocks* $\varphi^{(j)} : E^{(j)} \to E^{(j+1)}$ of size 2^f, and a last block of size 2^{f_0}, where $f_0 = e - (n - 1) \cdot f$. For every $\varphi^{(j)}$, the kernel $\langle K^{(j)} \rangle$ is given by the generator $K^{(j)} = P^{(j)} + [s^{(j)}]Q^{(j)}$ for a deterministic basis $(P^{(j)}, Q^{(j)})$ of $E^{(j)}[2^f]$.

In the first block, after sampling $(P^{(1)}, Q^{(1)})$ via FindBasis, the bit b indicates whether $P^{(1)}$ and $Q^{(1)}$ have to be swapped before running FindKernel. For the following blocks, the verifier pushes $Q^{(j)}$ through the isogeny $\varphi^{(j)}$ to get a point $Q^{(j+1)} \leftarrow \varphi^{(j)}(Q^{(j)})$ on $E^{(j+1)}$ of order 2^f above $(0, 0)$.[16] Hence, for $j > 1$ FindBasis only needs to find a suitable point $P^{(j)}$ to complete the basis $(P^{(j)}, Q^{(j)})$. Furthermore, $K^{(j)}$ is never above $(0, 0)$ for $j > 1$, which ensures cyclicity when composing $\varphi^{(j)}$ with $\varphi^{(i-1)}$. In all cases we use $s^{(j)}$ from σ to compute the kernel generator $K^{(j)}$ via FindKernel and $\varphi^{(j)}$ via ComputeIsogeny.

The last block of degree 2^{f_0} uses $Q^{(n)} \leftarrow [2^{f-f_0}]\varphi^{(n-1)}(Q^{(n-1)})$ and samples another point $P^{(n)}$ as basis of $E^{(n)}[2^{f_0}]$. In the following, we will often assume $f_0 = f$ for the sake of simplicity.[17] An algorithmic description of a single block of SQIsign (NIST) is given in Algorithm 2 in Appendix B.

[16] A point P is said to be *above* a point R if $[k]P = R$ for some $k \in \mathbb{N}$.

[17] In contrast to earlier versions, SQIsign (NIST) fixes $f_0 = f$. However, our analysis benefits from allowing $f_0 < f$.

Step (b). Computing $\widehat{\varphi_{\text{chall}}}$ requires a single isogeny of smooth degree $D_{\text{chall}} \approx 2^\lambda$. For the primes given in SQIsign (NIST), we have $E_2[D_{\text{chall}}] \subseteq E_2(\mathbb{F}_{p^2})$. Thus, we compute φ_{chall} by deterministically computing a basis (P, Q) for $E_2[D_{\text{chall}}]$ and finding the kernel $\langle K \rangle$ for $\widehat{\varphi_{\text{chall}}} : E_2 \to E_1$. For $f < \lambda$, we have $D_{\text{chall}} = 2^f \cdot 3^g$, and split this process into two parts.

Given the basis (P, Q) for $E_2[D_{\text{chall}}]$, we compute $(P_2, Q_2) = ([3^g]P, [3^g]Q)$ as basis of $E_2[2^f]$, and use $K_2 = P_2 + [s_2]Q_2$, where b_2 indicates whether P_2 and Q_2 have to be swapped prior to computing K_2. We compute $\varphi_2 : E_2 \to E_2'$ with kernel $\langle K_2 \rangle$, and $P_3 = [2^f]\varphi_2(P)$ and $Q_3 = [2^f]\varphi_3(Q)$ form a basis of $E_2'[3^g]$. Then b_3 indicates a potential swap of P_3 and Q_3, while $K_3 = P_3 + [s_3]Q_3$ is the kernel generator of the isogeny $\varphi_3 : E_2' \to E_1$. Thus, we have $\widehat{\varphi_{\text{chall}}} = \varphi_3 \circ \varphi_2$. If $f \geq \lambda$, we require only the first step.

We furthermore verify that the composition of φ_{resp} and $\widehat{\varphi_{\text{chall}}}$ is cyclic, by checking that the first 2-isogeny step of φ_2 does not revert the last 2-isogeny step of $\varphi^{(n)}$. This guarantees that $\widehat{\varphi_{\text{chall}}} \circ \varphi_{\text{resp}}$ is non-backtracking, hence cyclic.

Step (c). On E_1, the verifier uses the point $Q' \leftarrow \widehat{\varphi_{\text{chall}}}(Q')$, where Q' is some (deterministically generated) point, linearly independent from the generator of $\widehat{\varphi_{\text{chall}}}$, and r (given in σ) to compute $[r]Q'$, and checks if $[r]Q'$ matches the hashed point $K_{\text{chall}} = H(\texttt{msg}, E_1)$ with hash function H.

4.2 Impact of large f on verification

The techniques of Section 3 extend the possible range of f to any size below $\log(p)$. This gives two benefits to the cost of verification, especially when $f \geq \lambda$.

Number of blocks in φ_{resp}. The larger f is, the fewer blocks of size 2^f are performed in **Step (a)**. Per block, the dominating part of the cost are FindBasis and FindKernel as we first need to complete the torsion basis $(P^{(j)}, Q^{(j)})$ for $E^{(j)}[2^f]$ (given $Q^{(j)}$ if $j > 1$), followed by computing $K^{(j)} = P^{(j)} + [s^{(j)}]Q^{(j)}$. By minimizing the number of blocks n, we minimize the amount of times we perform FindBasis and FindKernel, and the cost of each individual FindKernel only mildly increases, as $s^{(j)}$ increases in size. The overall cost of ComputeIsogeny, that is, performing the n isogenies of degree 2^f given their kernels $K^{(j)}$, only moderately increases with growing f.

We further note that larger f requires fewer T-isogeny computations for the signer, hence signing performance also benefits from smaller n.

Challenge isogeny. When $f \geq \lambda$, we can simply set $D_{\text{chall}} = 2^\lambda$, which has two main benefits.

- The cost of FindBasis for this step is reduced as finding a basis for $E[2^\lambda]$ is much easier than a basis search for $E[2^f \cdot 3^g]$.

– The cost for ComputeIsogeny for φ_{chall} decreases as we only have to compute a chain of 2-isogenies instead of additional 3-isogenies.

4.3 Implementation and benchmark of cost in $\mathbb{F}_p$-multiplications

To measure the influence of the size of f on the performance, we implemented SQIsign verification for the NIST Level I security parameter set in Python, closely following SQIsign (NIST). As is standard in isogeny-based schemes, we use x-only arithmetic and represent points and curve coefficients projectively. The benchmark counts $\mathbb{F}_p$-operations and uses a cost metric that allows us to estimate the runtime of real-world implementations for 256-bit primes $p^{(f)}$, where $p^{(f)}$ denotes a prime such that 2^f divides $p^{(f)} + 1$. We benchmark primes $p^{(f)}$ for all values $50 \leq f \leq 250$. These results serve as a baseline to which we compare the optimisations that we introduce in Sections 5 and 6.

We briefly outline how SQIsign (NIST) implements the three subroutines FindBasis, FindKernel, and ComputeIsogeny.

FindBasis. We search for points of order 2^f by sampling x-coordinates in a specified order,[18] and check if the corresponding point P lies on E (and not on its twist E^t). We then compute $P \leftarrow [\frac{p+1}{2^f}]P$ and verify that $[2^{f-1}]P \neq \infty$. Given two points $P, Q \in E$ of order 2^f, we verify linear independence by checking that $[2^{f-1}]P \neq [2^{f-1}]Q$, and discard and re-sample the second point otherwise.

FindKernel. Given a basis (P, Q), FindKernel computes $K = P + [s]Q$ via the `3ptLadder` algorithm as used in SIKE [27]. In addition to the x-coordinates x_P and x_Q of P and Q, it requires the x-coordinate x_{P-Q} of $P - Q$. Hence, after running FindBasis, we further compute x_{P-Q} as described in SQIsign (NIST) [12].

ComputeIsogeny. Given a kernel generator K of order 2^f, ComputeIsogeny follows the approach of SIKE [27], and computes the 2^f-isogeny $\varphi^{(j)}$ as a chain of 4-isogenies for efficiency reasons. If f is odd, we further compute a single 2-isogeny. Following SQIsign (NIST), ComputeIsogeny proceeds as follows:

1. Compute $R = [2^{f-2}]K$ and the corresponding 4-isogeny φ with kernel $\langle R \rangle$. Note that the point $(0, 0)$ might be contained in $\langle R \rangle$ for the first block in φ_{resp}, which requires a special 4-isogeny formula. Thus, we check if this is the case and call the suitable 4-isogeny function. We set $K \leftarrow \varphi(K)$.
2. If f is odd, we compute $R = [2^{f-3}]K$, the 2-isogeny φ with kernel $\langle R \rangle$, and $K \leftarrow \varphi(K)$.

[18] SQIsign (NIST) fixes the sequence $x_k = 1 + k \cdot i$ with $i \in \mathbb{F}_{p^2}$ such that $i^2 = -1$ and picks the smallest k for which we find a suitable point.

3. Compute the remaining isogeny of degree $2^{f'}$ with even f' as a chain of 4-isogenies, where $(0,0)$ is guaranteed not to lie in any of the kernels.

In the last step, SQIsign (NIST) uses *optimal strategies* as in SIKE [27] to compute a chain of 4-isogenies. Naive multiplicative strategies would compute $R = [2^{f'-2j}]K$, the 4-isogeny φ with kernel $\langle R \rangle$, and $K \leftarrow \varphi(K)$ for $j = 1, \ldots, f'/2$. However, this strategy is dominated by costly doublings. Instead, we can save intermediate multiples of K during the computation of $R = [2^{f'-2j}]K$, and push them through isogenies to save multiplicative effort in following iterations. Optimal strategies that determine which multiples are pushed through isogenies and minimise the cost can be found efficiently [20, 27].

We note that for $f < \lambda$ the computation of $\widehat{\varphi_{\text{chall}}}$ requires small adaptations to these algorithms to allow for finding a basis of $E[D_{\text{chall}}]$ and computing 3-isogenies. Most notably, SQIsign (NIST) does *not* use optimised formulas or optimal strategies for 3-isogenies from SIKE [27], but uses a multiplicative strategy and general odd-degree isogeny formulas [16, 32]. We slightly deviate from SQIsign (NIST) by implementing optimised 3-isogeny formulas, but note that the performance difference is minor and in favor of SQIsign (NIST).

Cost metric. In implementations, $\mathbb{F}_{p^2}$-operations usually call underlying $\mathbb{F}_p$-operations. We follow this approach and use the total number of $\mathbb{F}_p$-operations in our benchmarks. As cost metric, we express these operations in terms of $\mathbb{F}_p$-multiplications, with $\mathbf{S} = 0.8 \cdot \mathbf{M}$, ignoring $\mathbb{F}_p$-additions and subtractions due to their small impact on performance. $\mathbb{F}_p$-inversions, $\mathbb{F}_p$-square roots, and Legendre symbols over $\mathbb{F}_p$ require exponentiations by an exponent in the range of p, hence we count their cost as $\log p$ $\mathbb{F}_p$-multiplications. In contrast to measuring clock cycles of an optimised implementation, our cost metric eliminates the dependence on the level of optimisation of finite field arithmetic and the specific device running SQIsign, hence, can be considered more general.

Benchmark results. Figure 2 shows the verification cost for the NIST Level I-sized primes $p^{(f)}$ for $50 \leq f \leq 250$, fixing $e = 975$, using our cost metric. For more efficient benchmarking, we sample random public key curves and signatures σ of the correct form instead of signatures generated by the SQIsign signing procedure.

The graph shows the improvement for $f \geq 128$. Furthermore, we can detect when the number of blocks n decreases solely from the graph (e.g. $f = 122, 140, 163, 195, 244$). The cost of sampling a 2^f-torsion basis is highly variable between different runs for the same prime, which is visible from the oscillations of the graph. The performance for odd f is worse in general due to the inefficient integration of the 2-isogeny, which explains the zigzag-shaped graph.

From the above observations, we conclude that $f \geq \lambda$ is significantly faster for verification, with local optima found at $f = 195$ and $f = 244$, due to those being (almost) exact divisors of the signing length $e = 975$.

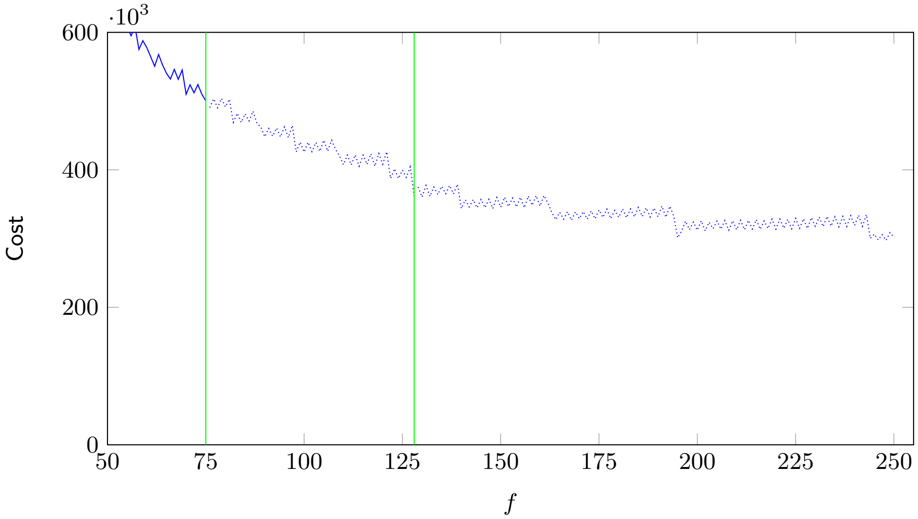


Fig. 2: Cost in $\mathbb{F}_p$-multiplications for verification at NIST Level I security, for varying f and $p^{(f)}$, averaged over 1024 runs per prime. The green vertical lines mark $f = 75$ as used in SQIsign (NIST) for signing without extension fields, and $f = \lambda = 128$, beyond which we can set $D_{\text{chall}} = 2^\lambda$. The dotted graph beyond $f = 75$ is only accessible when signing with extension fields.

Remark 6. The average cost of FindBasis differs significantly between primes p even if they share the same 2^f-torsion. This happens because SQIsign (NIST) finds basis points from a pre-determined sequence $[x_1, x_2, x_3, \dots]$ with $x_j \in \mathbb{F}_{p^2}$. As we will see in Section 5, these x_j values can not be considered random: some values x_j are certain to be above a point of order 2^f, while others are certain not to be, for any supersingular curve over p.

5 Optimisations for verification

In this section, we show how the improvements from Section 3 that increase f beyond λ together with the analysis in Section 4 allow several other optimisations that improve the verification time of SQIsign in practice. Whereas the techniques in Section 3 allow us to decrease the *number* of blocks, in this section, we focus on the operations occurring *within blocks*. We optimise the cost of FindBasis, FindKernel and ComputeIsogeny.

We first analyse the properties of points that have full 2^f-torsion, and use these properties to improve FindBasis and FindKernel for general f. We then describe several techniques specifically for $f \geq \lambda$. Altogether, these optimisations significantly change the implementation of verification in comparison to SQIsign (NIST). We remark that the implementation of the signing procedure must be altered accordingly, as exhibited by our implementation.

Notation. As we mostly focus on the subroutines *within* a specific block $E^{(j)} \to E^{(j+1)}$, we will omit the superscripts in $E^{(j)}, K^{(j)}, P^{(j)}, \dots$ and write $E, K, P, \dots$ to simplify notation.

For reference throughout this section, the pseudocode for a single block in the verification procedure of SQIsign (NIST) and of our optimised variant is in Appendix B as Algorithm 2 and Algorithm 3, respectively.

5.1 Basis generation for full 2-power torsion

We first give a general result on points having full 2^f-torsion that we will use throughout this section. This theorem generalises previous results [17, 31] and will set the scene for easier and more efficient basis generation for $E[2^f]$.

Theorem 2. *Let $E : y^2 = (x - \lambda_1)(x - \lambda_2)(x - \lambda_3)$ be an elliptic curve over $\mathbb{F}_{p^2}$ with $E[2^f] \subseteq E(\mathbb{F}_{p^2})$ the full 2-power torsion. Let $L_i = (\lambda_i, 0)$ denote the points of order 2 and $[2]E$ denote the image of E under $[2] : P \mapsto P + P$ so that $E \setminus [2]E$ are the points with full 2^f-torsion. Then*

$$Q \in [2]E \textit{ if and only if } x_Q - \lambda_i \textit{ is square for } i = 1, 2, 3.$$

More specifically, for $Q \in E \setminus [2]E$, Q is above L_i if and only if $x_Q - \lambda_i$ is square and $x_Q - \lambda_j$ is non-square for $j \neq i$.

Proof. It is well-known that $Q = (x, y) \in [2]E$ if and only if $x - \lambda_1$, $x - \lambda_2$ and $x - \lambda_3$ are all three squares [26, Ch. 1, Thm. 4.1]. Thus, for $Q \in E \setminus [2]E$, one of these three values must be a square, and the others non-squares (as their product must be y^2, hence square). We proceed similarly as the proof of [31, Thm. 3]. Namely, let P_1, P_2 and P_3 denote points of order 2^f above $L_1 = (\lambda_1, 0)$, $L_2 = (\lambda_2, 0)$ and $L_3 = (\lambda_3, 0)$, respectively, of order 2. A point $Q \in E \setminus [2]E$ must lie above one of the L_i. Therefore, the reduced Tate pairing of degree 2^f of P_i and Q gives a primitive 2^f-th root of unity if and only if Q is not above L_i. Let $\zeta_i = e_{2^f}(P_i, Q)$, then by [25, Thm. IX.9] we have

$$\zeta_i^{2^{f-1}} = e_2(L_i, Q).$$

We can compute $e_2(L_i, Q)$ by evaluating a Miller function f_{2,L_i} in Q, where $\operatorname{div} f_{2,L_i} = 2(L_i) - 2(\mathcal{O})$. The simplest option is the line that doubles L_i, that is, $f_{2,L_i}(x, y) = x - \lambda_i$, hence

$$e_2(L_i, Q) = (x_Q - \lambda_i)^{\frac{p^2-1}{2}}.$$

Applying Euler's criterion to this last term, we get that if $x_Q - \lambda_i$ is square, then ζ_i is not a primitive 2^f-th root and hence Q must be above L_i, whereas if $x_Q - \lambda_i$ is non-square, then ζ_i is a primitive 2^f-th root and hence Q is not above L_i. □

Note that for Montgomery curves $y^2 = x^3 + Ax^2 + x = x(x - \alpha)(x - 1/\alpha)$, the theorem above tells us that non-squareness of x_Q for $Q \in E(\mathbb{F}_{p^2})$ is enough to imply Q has full 2^f-torsion and is not above $(0, 0)$ [31, Thm. 3].

Finding points with $\mathbf{2^f}$-torsion above $\mathbf{(0, 0)}$. We describe two methods to efficiently sample Q above $(0, 0)$, based on Theorem 2.

1. **Direct x sampling.** By deterministically sampling $x_Q \in \mathbb{F}_p$, we ensure that x_Q is square in $\mathbb{F}_{p^2}$. Hence, if Q lies on E and $x_Q - \alpha \in \mathbb{F}_{p^2}$ is non-square, where α is a root of $x^2 + Ax + 1$, then Theorem 2 ensures that $Q \in E \setminus [2]E$ and above $(0, 0)$.
2. **Smart x sampling.** We can improve this using the fact that α is always square [3, 15]. Hence, if we find $z \in \mathbb{F}_{p^2}$ such that z is square and $z - 1$ is non-square, we can choose $x_Q = z\alpha$ square and in turn $x_Q - \alpha = (z - 1)\alpha$ non-square. Again, by Theorem 2 if Q is on E, this ensures Q is above $(0, 0)$ and contains full 2^f-torsion. Hence, we prepare a list $[z_1, z_2, \ldots]$ of such values z for a given prime, and try $x_j = z_j\alpha$ until x_j is on E.

Both methods require computing α, dominated by one $\mathbb{F}_{p^2}$-square root. Direct sampling computes a Legendre symbol of $x^3 + Ax^2 + x$ per x to check if the corresponding point lies on E. If so, we check if $x - \alpha$ is non-square via the Legendre symbol. On average, this requires four samplings of x and six Legendre symbols to find a suitable x_Q with $Q \in E(\mathbb{F}_{p^2})$, and, given that we can choose x_Q to be small, we can use fast scalar multiplication on x_Q (see Appendix A).

In addition to computing α, smart sampling requires the Legendre symbol computation of $x^3 + Ax^2 + x$ per x. On average, we require two samplings of an x to find a suitable x_Q, hence saving four Legendre symbols in comparison to direct sampling. However, we can no longer choose x_Q small, which means that improved scalar multiplication for small x_Q is not available.

Finding points with $\mathbf{2^f}$-torsion *not* above $\mathbf{(0, 0)}$. As shown in [31], we find a point P with full 2^f-torsion *not* above $(0, 0)$ by selecting a point on the curve with non-square x-coordinate. Non-squareness depends only on p, not on E, so a list of small non-square values can be precomputed. In this way, finding such a point P simply becomes finding the first value x_P in this list such that the point $(x_P, -)$ lies on $E(\mathbb{F}_{p^2})$, that is, $x_P^3 + Ax_P^2 + x_P$ is square. On average, this requires two samplings of x, hence two Legendre symbol computations.

5.2 General improvements to verification

In this section, we describe improvements to SQIsign verification and present new optimisations, decreasing the cost of the three main subroutines of verification.

Known techniques from literature. There are several state-of-the-art techniques in the literature on efficient implementations of elliptic curve or isogeny-based schemes that allow for general improvements to verification, but are not included in SQIsign (NIST). We implemented such methods, e.g., to improve scalar multiplication $P \mapsto [n]P$ and square roots. The details are described in Appendix A. In particular, we use that $P \mapsto [n]P$ is faster when x_P is small.

Improving the subroutine FindBasis. In SQIsign (NIST), to find a complete basis for $E[2^f]$ we are given a point $Q \in E[2^f]$ lying above $(0,0)$ and need to find another point $P \in E(\mathbb{F}_{p^2})$ of order 2^f not lying above $(0,0)$. We sample P directly using x_P non-square, as described above and demonstrated by [31], and in particular can choose x_P small. We then compute $P \leftarrow [\frac{p+1}{2^f}]P$ via fast scalar multiplication to complete the torsion basis (P,Q).

Improved strategies for ComputeIsogeny. Recall that ComputeIsogeny follows three steps in SQIsign (NIST): it first computes a 4-isogeny that may contain $(0,0)$ in the kernel, and a 2-isogeny if f is odd, before entering an optimal strategy for computing the remaining chain of 4-isogenies. However, the first two steps include many costly doublings. We improve this by adding these first two steps in the optimal strategy. If f is even, this is straightforward, with a simple check for $(0,0)$ in the kernel in the first step. For odd f, we add the additional 2-isogeny in this first step.[19] For simplicity of the implementation, we determine optimal strategies as in SIKE [27], thus we assume that only 4-isogenies are used.

Note that techniques for strategies with variable isogeny degrees are available from the literature on CSIDH implementations [13]. However, the performance difference is very small, hence our simplified approach appears to be preferable.

In addition to optimising 4-isogeny chains, we implemented optimised 3-isogeny chains from SIKE [27] for the computation of $\widehat{\varphi_{\text{chall}}}$ when $f < 128$.

5.3 To push, or not to push[20]

In SQIsign (NIST), the point Q is pushed through φ so that we easily get the basis point above $(0,0)$ on the image curve, and we can then use Theorem 2 to sample the second basis point P. Instead of pushing Q, one can also use Theorem 2 to efficiently sample this basis point Q above $(0,0)$. Although pushing Q seems very efficient, for larger f we are pushing Q through increasingly larger isogeny chains, whereas sampling becomes increasingly more efficient as multiplication cost by $\frac{p+1}{2^f}$ decreases. Furthermore, sampling

[19] In particular, we compute $R' = [2^{f-3}]K$ and $R = [2]R'$, a 4-isogeny with kernel $\langle R \rangle$, push R' through, and compute a 2-isogeny with kernel $\langle R' \rangle$.

[20] –that is, the $\boldsymbol{Q}$.

both P and Q allows us to use those points as an *implicit basis* for $E[2^f]$, even if their orders are multiples of 2^f, as described in more detail below. We observe experimentally that this makes sampling Q, instead of pushing Q, more efficient for $f > 128$.

Using implicit bases. Using Theorem 2, it is possible to find points P and Q efficiently so that both have full 2^f-torsion. The pair (P, Q) is not an *explicit basis* for $E[2^f]$, as the orders of these points are likely to be multiples of 2^f. However, instead of multiplying both points by the cofactor to find an explicit basis, we can use these points implicitly, as if they were a basis for $E[2^f]$. This allows us to compute $K = P + [s]Q$ first, and only then multiply K by the cofactor. This saves a full scalar multiplication by the cofactor $\frac{p+1}{2^f}$. We refer to such a pair (P, Q) as an *implicit basis* of $E[2^f]$. Algorithmically, implicit bases combine FindBasis and FindKernel into a single routine FindBasisAndKernel.

5.4 Improved challenge for $f \geq \lambda$

Recall from Section 4.2 that when $f \geq \lambda$, we can simply set $D_{\text{chall}} = 2^\lambda$. This decreases the cost of FindBasis for the challenge computation considerably, as we can now use Theorem 2 to find a basis for $E[2^\lambda]$.

Improving FindBasis for the challenge isogeny when $f \geq \lambda$. We use Theorem 2 twice, first to find P not above $(0, 0)$ having full 2^f-torsion and then to find Q above $(0, 0)$ having full 2^f-torsion. We choose x_P and x_Q small such that faster scalar multiplication is available. We find the basis for $E[2^\lambda]$ by $P \leftarrow [\frac{p+1}{2^f}]P$ followed by $f - \lambda$ doublings, and $Q \leftarrow [\frac{p+1}{2^f}]Q$ followed by $f - \lambda$ doublings.[21] Alternatively, if Q is pushed through isogenies, we can reuse $Q \leftarrow \varphi^{(n)}(Q^{(n)}) \in E[2^f]$ from the computation of the last step of φ_{resp}, so that we get a basis point for $E[2^\lambda]$ by $f - \lambda$ doublings of Q. Reusing this point Q also guarantees cyclicity of $\widehat{\varphi_{\text{chall}}} \circ \varphi_{\text{resp}}$.

Remark 7. For SQIsign without extension fields, obtaining $f \geq \lambda$ seems infeasible, hence the degree D of φ_{chall} is $2^f \cdot 3^g$. Nevertheless, some optimizations are possible in the computation of φ_{chall} in this case. FindBasis for $E[2^f \cdot 3^g]$ benefits from similar techniques as previously used in SIDH/SIKE, as we can apply known methods to improve generating a torsion basis for $E[3^g]$ coming from 3-descent [17, § 3.3]. Such methods are an analogue to generating a basis for $E[2^f]$ as described in Theorem 2 and [31, Thm. 3].

6 Size-speed trade-offs in SQIsign signatures

The increase in f also enables several size-speed trade-offs by adding further information in the signature or by using uncompressed signatures. Some trade-offs were already present

[21] Algorithmically, this is faster than a single scalar multiplication by $2^{f-\lambda} \cdot \frac{p+1}{2^f}$.

in earlier versions of SQIsign [21], however, by using large f and the improvements from Section 5, they become especially worthwhile.

We take a slightly different stance from previous work on SQIsign as for many use cases the main road block to using SQIsign is the efficiency of verification in cycles. In contrast, in several applications the precise size of a signature is less important as long as it is below a certain threshold.[22] For example, many applications can handle the combined public key and signature size of RSA-2048 of 528 bytes, while SQIsign (NIST) features a combined size of only 241 bytes. In this section, we take the 528 bytes of RSA-2048 as a baseline, and explore size-speed trade-offs for SQIsign verification with data sizes up to this range.

We note that the larger signatures in this section encode the same information as standard SQIsign signatures, hence have no impact on the security.

6.1 Adding seeds for the torsion basis in the signature

We revisit an idea that was previously present in SQIsign verification [21] (but no longer in [12] or [22]), and highlight its particular merits whenever $f \geq \lambda$, as enabled by signing with extension fields. So far, we have assumed that completing or sampling a basis for $E[2^f]$ is done by deterministically sampling points. Recall from Section 5.1 that sampling x_P resp. x_Q (when not pushing Q) on average requires the computation of several Legendre symbols resp. square roots. We instead suggest using a seed to find x_P (when pushing Q) or x_P and x_Q (otherwise), which we include in the signature, so that the verifier saves all of the above cost for finding x_P, resp. x_Q. Finding these seeds adds negligible overhead for the signer, while verification performance improves. Signer and verifier are assumed to agree upon all precomputed values.

Seeding a point *not* above $(0,0)$. For x_P *not* above $(0,0)$, we fix a large enough $k > 0$ and precompute the 2^k smallest values $u_j \in \mathbb{F}_p$ such that $u_j + i \in \mathbb{F}_{p^2}$ is non-square (where i is the same as in Section 5). During signing, we pick the smallest u_j such that $x_P = u_j + i$ is the x-coordinate of a point $P \in E(\mathbb{F}_{p^2})$, and add the index j to the signature as a seed for x_P. Theorem 2 ensures that any $P \in E(\mathbb{F}_{p^2})$ for non-square x_P is a point with full 2^f-torsion not above $(0,0)$. This furthermore has the advantage of fast scalar multiplication for x_P as the x-coordinate is very small.

Seeding a point *above* $(0,0)$. As noted above, when f is large, it is faster to deterministically compute a point of order 2^f above $(0,0)$ than to push Q through φ. We propose a similar seed here for fixed large enough $k > 0$, using Theorem 2 and the "direct sampling" approach from Section 5.1. During signing, we pick the smallest $j \leq 2^k$ such that $x_Q = j$

[22] See https://blog.cloudflare.com/sizing-up-post-quantum-signatures/.

is the x-coordinate of a point $Q \in E(\mathbb{F}_{p^2})$ and $x_Q - \alpha$ is non-square. We add $x_Q = j$ to the signature as seed.

Note that when using both seeding techniques, we do not explicitly compute $[\frac{p+1}{2^f}]P$ or $[\frac{p+1}{2^f}]Q$, but rather use the seeded points P and Q as an implicit basis, as described in Section 5.3.

Size of seeds. Per seeded point, we add k bits to the signature size. Thus, we must balance k such that signatures are not becoming too large, while failure probabilities for not finding a suitable seed are small enough. In particular, seeding x_P resp. x_Q via direct sampling has a failure probability of $\frac{1}{2}$ resp. $\frac{3}{4}$ per precomputed value. For the sake of simplicity, we set $k = 8$ for both seeds, such that every seed can be encoded as a separate byte.[23] This means that the failure rate for seeding Q is $(\frac{3}{4})^{256} \approx 2^{-106.25}$ for our choice, while for P it is 2^{-256}. Theoretically it is still possible that seeding failures occur. In such a case, we simply recompute KLPT. We furthermore include similar seeds for the torsion basis on E_A and E_2, giving a size increase of $(n+1) \cdot 2$ bytes.

The synergy with large f now becomes apparent. The larger f gets, the fewer blocks n are required, hence adding fewer seeds overall. For $f = 75$, the seeds require an additional 28 bytes when seeding both P and Q. For $f = 122, 140, 163, 195, 244$ this drops to 18, 16, 14, 12, and 10 additional bytes, respectively, to the overall signature size of 177 bytes for NIST Level I security.

Remark 8. Instead of using direct sampling for Q with failure probability $\frac{3}{4}$, we can reduce it to $\frac{1}{2}$ via "smart sampling" (see Section 5.1). However, this requires the verifier to compute α via a square root to set $x_Q = z\alpha$ with seeded z. We thus prefer direct sampling for seeded Q, which incurs no such extra cost.

6.2 Uncompressed signatures

In cases where f is very large, and hence the number of blocks is small, in certain cases it is worthwhile to replace the value s in the signature by the full x-coordinate of $K = P + [s]Q$. In essence, this is the uncompressed version of the SQIsign signature σ, and we thus refer to this variant as *uncompressed* SQIsign.

Speed of uncompressed signatures. Adding the precise kernel point K removes the need for both FindBasis and FindKernel, leaving ComputeIsogeny as the sole remaining cost. This speed-up is significant, and leaves little room for improvement beyond optimizing the cost of computing isogenies. The cost of verification in this case is relatively constant, as computing an 2^e-isogeny given the kernels is only slightly affected by the size of f,

[23] Note that for equal failing rates the number of possible seeds for P can be chosen smaller than for Q, hence slightly decreasing the additional data sizes.

as is visible in the black dashed line in Figure 3. This makes uncompressed SQIsign an attractive alternative in cases where the signature size, up to a certain bound, is less relevant.

Size of uncompressed signatures. Per step, this increases the size from $\log(s) \approx f$ to $2 \cdot \log(p)$ bits, which is still relatively size efficient when f is close to $\log(p)$. For recomputing φ_{chall}, we take a slightly different approach than before. We add the Montgomery coefficient of E_1 to the signature, and seeds for a basis of $E[2^f]$. From this, the verifier can compute the kernel generator of φ_{chall}, and verify that the j-invariant of its codomain matches E_2. Hence this adds $2 \cdot \log(p)$ bits for E_1 and two bytes for seeds to the signature, for a total of $(n+1) \cdot (\log p/4) + 2$ bytes.

For $f = 244$, this approach less than doubles the signature size from 177 bytes to 322 bytes for NIST Level I security, for $f = 145$, the signature becomes approximately 514 bytes, while for the current NIST Level I prime with $f = 75$, the size would become 898 bytes. When adding the public key size of 64 bytes, especially the first two cases still appear to be reasonable alternatives to RSA-2048's combined data size of 528 bytes.

Remark 9. Uncompressed signatures significantly simplify verification, as many functionalities required for compressed signatures are not necessary. Hence, this allows for a much more compact code base, which might be important for use cases featuring embedded devices with strict memory requirements.

7 Primes and Performance

In this section we show the performance of verification for varying f, using the optimisations from the previous sections. Further, we find specific primes with suitable f for $n = 4$ and $n = 7$, and report their signing performance using our SageMath implementation, comparing it with the current SQIsign (NIST) prime.

7.1 Performance of optimised verification

To compare the verification performance of our optimised variants with compressed signatures to SQIsign (NIST) and SQIsign (LWXZ),[24] we run benchmarks in the same setting as in Section 4.3. In particular, Figure 3 shows the cost of verification for the NIST Level I primes $p^{(f)}$ for $50 \leq f \leq 250$. As before, we sample random public key curves and signatures σ of the correct form instead of using signatures generated by the SQIsign signing procedure.

[24] Our implementation of SQIsign (LWXZ) [31] is identical to SQIsign (NIST) except for the improved sampling of P described in Section 5.1.

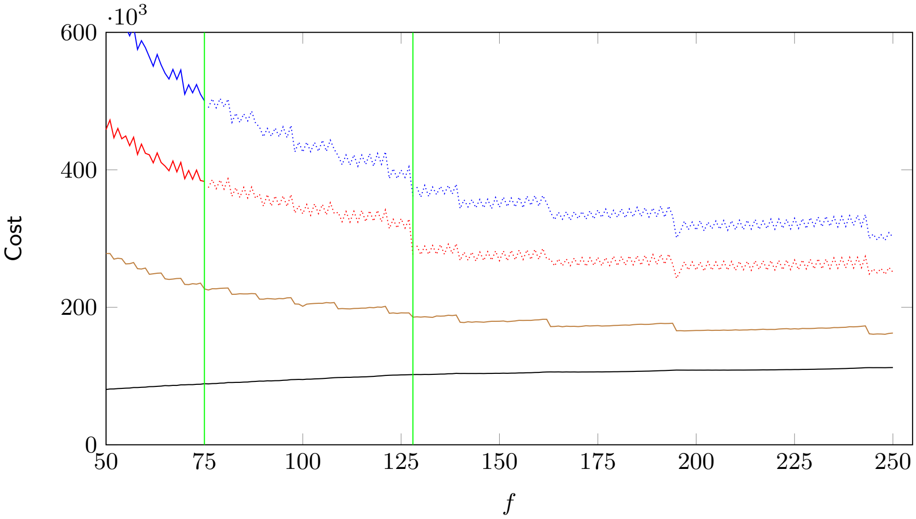


Fig. 3: Extended version of Figure 2 showing the cost in $\mathbb{F}_p$-multiplications for verification at NIST Level I security, for varying f and $p^{(f)}$, averaged over 1024 runs per prime. In addition to SQIsign (NIST) in blue, it shows the performance of SQIsign (LWXZ) in red, our fastest compressed `ApresSQI` variant in brown, and uncompressed `ApresSQI` in black.

For the sake of simplicity, Figure 3 displays only the fastest compressed `ApresSQI` variant, namely the version that does not push Q through isogenies and uses seeds to sample P and Q. This variant significantly outperforms both SQIsign (NIST) and SQIsign (LWXZ) already at $f = 75$, at the cost of slightly larger signatures. A detailed description and comparison of all four compressed variants is in Appendix C, which shows that our unseeded variants achieve similar large speed-ups with no increase in signature size. Lastly, the uncompressed variant achieves the fastest speed, although at a significant increase in signature size.

7.2 Finding specific primes

We now give two example primes, one prime optimal for 4-block verification, as well as the best we found for 7-block verification. The "quality" of a prime p is measured using the cost metric $\textsc{SigningCost}_p$ defined in Section 3.3.

Optimal 4-block primes. For 4-block primes, taking $e = 975$ as a baseline, we need f bigger than 244. In other words, we are searching for primes of the form

$$p = 2^{244}N - 1$$

where $N \in [2^4, 2^{12}]$ (accepting primes between 250 and 256 bits). This search space is quickly exhausted. For each prime of this form, we find the optimal torsion T to use, minimising $\textsc{SigningCost}_p(T)$. The prime with the lowest total cost in this metric, which we denote p_4, is

$$p_4 = 2^{246} \cdot 3 \cdot 67 - 1$$

Balanced primes. Additionally, we look for primes that get above the significant $f > 128$ line, while minimizing $\textsc{SigningCost}_p(T)$. To do this, we adopt the "sieve-and-boost" technique used to find the current SQIsign primes [12, §5.2.1]. However, instead of looking for divisors of $p^2 - 1$, we follow Theorem 1 and look for divisors of

$$\prod_{n=1}^{k} \Phi_n(p^2)/2$$

to find a list of good candidate primes. This list is then sorted with respect to their signing cost according to $\textsc{SigningCost}_p$. The prime with the lowest signing cost we could find, which we call p_7, is

$$p_7 = 2^{145} \cdot 3^9 \cdot 59^3 \cdot 311^3 \cdot 317^3 \cdot 503^3 - 1.$$

Remark 10. This method of searching for primes is optimised for looking for divisors of $p^2 - 1$, hence it might be suboptimal in the case of allowing torsion in higher extension fields. We leave it as future work to find methods which further take advantage of added flexibility in the prime search.

7.3 Performance for specific primes

We now compare the performance of the specific primes p_4, p_7, as well as the current NIST Level I prime p_{1973} used in SQIsign (NIST).

Signing performance. We give a summary of the estimated signing costs in Table 1. For p_{1973}, we include the metric "Adjusted Cost", which we compute as $\textsc{SigningCost}$ with the isogeny computations scaling as $\sqrt{\ell}\log\ell$ to (rather optimistically) account for the benefit of $\sqrt{\text{é}}$lu. Further, we ran our proof-of-concept SageMath implementation on the three primes, using SageMath 9.8, on a laptop with an Intel-Core i5-1038NG7 processor, averaged over five runs. An optimised C implementation will be orders of magnitude faster; we use these timings simply for comparison.

We note that the $\textsc{SigningCost}$-metric correctly predicts the ordering of the primes, though the performance difference is smaller than predicted. A possible explanation for this is that the $\textsc{SigningCost}$-metric ignores all overhead, such as quaternion operations, which roughly adds similar amounts of cost per prime.

Table 1: Comparison between estimated cost of signing for three different primes.

p	largest $\ell \mid T$	largest $\mathbb{F}_{p^{2k}}$	$\textsc{SigningCost}_p(T)$	Adj. Cost	Timing
p_{1973}	1973	$k = 1$	8371.7	1956.5	11m, 32s
p_7	997	$k = 23$	4137.9	-	9m, 20s
p_4	2293	$k = 53$	9632.7	-	15m, 52s

Our implementation uses $\sqrt{\text{élu}}$ whenever the kernel generator is defined over $\mathbb{F}_{p^2}$ and ℓ is bigger than a certain crossover point. This mainly benefits p_{1973}, as this prime only uses kernel generators defined over $\mathbb{F}_{p^2}$. The crossover point is experimentally found to be around $\ell > 300$ in our implementation, which is not optimal, compared to an optimised C implementation.[25] Nevertheless, we believe that these timings, together with the cost metrics, provide sufficient evidence that extension field signing in an optimised implementation stays in the same order of magnitude for signing time as staying over $\mathbb{F}_{p^2}$.

Verification performance. In Table 2, we summarise the performance of verification for p_{1973}, p_7, and p_4, both in terms of speed, and signature sizes.

Two highlights of this work lie in using p_7, both with and without seeds, having (almost) the same signature sizes as the current SQIsign signatures, but achieving a speed-up of factor 2.37 resp. 2.80 in comparison to SQIsign (NIST) and 1.82 resp. 2.15 in comparison to SQIsign (LWXZ), using p_{1973}. Another interesting alternative is using uncompressed p_4, at the cost of roughly double signature sizes, giving a speed-up of factor 4.46 in comparison to SQIsign (NIST) and 3.41 in comparison to SQIsign (LWXZ).

Remark 11. We analyse and optimise the cost of verification with respect to $\mathbb{F}_p$-operations. However, primes of the form $p = 2^f \cdot c - 1$ are considered to be particularly suitable for fast optimised finite field arithmetic, especially when f is large [4]. Hence, we expect primes like p_4 to improve significantly more in comparison to p_{1973} in low-level field arithmetic, leading to a larger speed-up than predicted in Table 2. Furthermore, other low-level improvements, such as fast non-constant time GCD for inversions or Legendre symbols, will improve the performance of primes in terms of cycles, which is unaccounted for by our cost metric.

Acknowledgements. We thank Craig Costello for helpful suggestions and comments on an earlier version of this work. We thank the anonymous Eurocrypt 2024 reviewers for their constructive feedback.

[25] For instance, work by Adj, Chi-Domínguez, and Rodríguez-Henríquez [1] gives the crossover point at $\ell > 89$, although for isogenies defined over $\mathbb{F}_p$.

Table 2: Comparison between verification cost for different variants and different primes, with cost given in terms of 10^3 $\mathbb{F}_p$-multiplications, using $\mathbf{S} = 0.8 \cdot \mathbf{M}$.

p	f	Implementation	Variant	Verif. cost	Sig. size
p_{1973}	75	SQIsign (NIST) [12]	-	500.4	177 B
		SQIsign (LWXZ) [31]	-	383.1	177 B
		`AprèsSQI`	unseeded	276.1	177 B
		`AprèsSQI`	seeded	226.8	195 B
p_7	145	`AprèsSQI`	unseeded	211.0	177 B
		`AprèsSQI`	seeded	178.6	193 B
		`AprèsSQI`	uncompressed	103.7	514 B
p_4	246	`AprèsSQI`	unseeded	185.2	177 B
		`AprèsSQI`	seeded	160.8	187 B
		`AprèsSQI`	uncompressed	112.2	322 B

References

[1] Gora Adj, Jesús-Javier Chi-Domínguez, and Francisco Rodríguez-Henríquez. "Karatsuba-based square-root Vélu's formulas applied to two isogeny-based protocols". In: *J. Cryptogr. Eng.* 13.1 (2023), pp. 89–106. DOI: 10.1007/S13389-022-00293-Y. URL: https://doi.org/10.1007/s13389-022-00293-y.

[2] Tom M. Apostol. "Resultants of cyclotomic polynomials". In: *Proceedings of the American Mathematical Society* 24.3 (1970), pp. 457–462.

[3] Roland Auer and Jaap Top. "Legendre Elliptic Curves over Finite Fields". In: *Journal of Number Theory* 95.2 (2002), pp. 303–312. ISSN: 0022-314X. DOI: https://doi.org/10.1006/jnth.2001.2760. URL: https://www.sciencedirect.com/science/article/pii/S0022314X0192760X.

[4] Jean-Claude Bajard and Sylvain Duquesne. "Montgomery-friendly primes and applications to cryptography". In: *J. Cryptogr. Eng.* 11.4 (2021), pp. 399–415. DOI: 10.1007/s13389-021-00260-z. URL: https://doi.org/10.1007/s13389-021-00260-z.

[5] Gustavo Banegas, Daniel J. Bernstein, Fabio Campos, Tung Chou, Tanja Lange, Michael Meyer, Benjamin Smith, and Jana Sotáková. "CTIDH: faster constant-time CSIDH". In: *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2021.4 (2021), pp. 351–387. DOI: 10.46586/tches.v2021.i4.351-387. URL: https://doi.org/10.46586/tches.v2021.i4.351-387.

[6] Gustavo Banegas, Valerie Gilchrist, Anaëlle Le Dévéhat, and Benjamin Smith. "Fast and Frobenius: Rational Isogeny Evaluation over Finite Fields". In: *LATINCRYPT 2023 - 8th International Conference on Cryptology and Information Security in Latin America.* Springer. 2023, pp. 129–148.

[7] Daniel J. Bernstein. *Differential addition chains.* 2006. URL: http://cr.yp.to/ecdh/diffchain-20060219.pdf.

[8] Daniel J. Bernstein, Luca De Feo, Antonin Leroux, and Benjamin Smith. "Faster computation of isogenies of large prime degree". In: *Open Book Series* 4.1 (2020), pp. 39–55.

[9] Jean-François Biasse, David Jao, and Anirudh Sankar. "A quantum algorithm for computing isogenies between supersingular elliptic curves". In: *International Conference on Cryptology in India.* Springer. 2014, pp. 428–442.

[10] Giacomo Bruno, Maria Corte-Real Santos, Craig Costello, Jonathan Komada Eriksen, Michael Meyer, Michael Naehrig, and Bruno Sterner. "Cryptographic Smooth Neighbors". In: *IACR Cryptol. ePrint Arch.* (2022), p. 1439. URL: https://eprint.iacr.org/2022/1439.

[11] Daniel Cervantes-Vázquez, Mathilde Chenu, Jesús-Javier Chi-Domínguez, Luca De Feo, Francisco Rodríguez-Henríquez, and Benjamin Smith. "Stronger and Faster Side-Channel Protections for CSIDH". In: *Progress in Cryptology - LATINCRYPT 2019 - 6th International Conference on Cryptology and Information Security in Latin America, Santiago de Chile, Chile, October 2-4, 2019, Proceedings.* Ed. by Peter Schwabe and Nicolas Thériault. Vol. 11774. Lecture Notes in Computer Science. Springer, 2019, pp. 173–193. DOI: 10.1007/978-3-030-30530-7_9. URL: https://doi.org/10.1007/978-3-030-30530-7_9.

[12] Jorge Chavez-Saab, Maria Corte-Real Santos, Luca De Feo, Jonathan Komada Eriksen, Basil Hess, David Kohel, Antonin Leroux, Patrick Longa, Michael Meyer, Lorenz Panny, Sikhar Patranabis, Christophe Petit, Francisco Rodríguez-Henríquez, Sina Schaeffler, and Benjamin Wesolowski. *SQIsign: Algorithm specifications and supporting documentation.* National Institute of Standards and Technology. 2023. URL: https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/spec-files/sqisign-spec-web.pdf.

[13] Jesús-Javier Chi-Domínguez and Francisco Rodríguez-Henríquez. "Optimal strategies for CSIDH". In: *Adv. Math. Commun.* 16.2 (2022), pp. 383–411. DOI: 10.3934/amc.2020116. URL: https://doi.org/10.3934/amc.2020116.

[14] Craig Costello. "B-SIDH: Supersingular Isogeny Diffie-Hellman Using Twisted Torsion". In: *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II.* Ed. by Shiho Moriai and Huaxiong Wang. Vol. 12492. Lecture Notes in Computer Science. Springer, 2020, pp. 440–463. DOI: 10.1007/978-3-030-64834-3_15. URL: https://doi.org/10.1007/978-3-030-64834-3_15.

[15] Craig Costello. "Computing Supersingular Isogenies on Kummer Surfaces". In: *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part III.* Ed. by Thomas Peyrin and

Steven D. Galbraith. Vol. 11274. Lecture Notes in Computer Science. Springer, 2018, pp. 428–456. DOI: 10.1007/978-3-030-03332-3_16. URL: https://doi.org/10.1007/978-3-030-03332-3_16.

[16] Craig Costello and Hüseyin Hisil. "A Simple and Compact Algorithm for SIDH with Arbitrary Degree Isogenies". In: *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*. Ed. by Tsuyoshi Takagi and Thomas Peyrin. Vol. 10625. Lecture Notes in Computer Science. Springer, 2017, pp. 303–329. DOI: 10.1007/978-3-319-70697-9_11. URL: https://doi.org/10.1007/978-3-319-70697-9_11.

[17] Craig Costello, David Jao, Patrick Longa, Michael Naehrig, Joost Renes, and David Urbanik. "Efficient Compression of SIDH Public Keys". In: *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part I*. Ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Vol. 10210. Lecture Notes in Computer Science. 2017, pp. 679–706. DOI: 10.1007/978-3-319-56620-7_24. URL: https://doi.org/10.1007/978-3-319-56620-7_24.

[18] Craig Costello, Michael Meyer, and Michael Naehrig. "Sieving for Twin Smooth Integers with Solutions to the Prouhet-Tarry-Escott Problem". In: *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part I*. Ed. by Anne Canteaut and François-Xavier Standaert. Vol. 12696. Lecture Notes in Computer Science. Springer, 2021, pp. 272–301. DOI: 10.1007/978-3-030-77870-5_10. URL: https://doi.org/10.1007/978-3-030-77870-5_10.

[19] Pierrick Dartois, Antonin Leroux, Damien Robert, and Benjamin Wesolowski. "SQISignHD: New Dimensions in Cryptography". In: *IACR Cryptol. ePrint Arch.* (2023), p. 436. URL: https://eprint.iacr.org/2023/436.

[20] Luca De Feo, David Jao, and Jérôme Plût. "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies". In: *J. Math. Cryptol.* 8.3 (2014), pp. 209–247. DOI: 10.1515/jmc-2012-0015. URL: https://doi.org/10.1515/jmc-2012-0015.

[21] Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. "SQISign: Compact Post-quantum Signatures from Quaternions and Isogenies". In: *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I*. Ed. by Shiho Moriai and Huaxiong Wang. Vol. 12491. Lecture Notes in Computer Science. Springer, 2020, pp. 64–93. DOI: 10.1007/978-3-030-64837-4_3. URL: https://doi.org/10.1007/978-3-030-64837-4_3.

[22] Luca De Feo, Antonin Leroux, Patrick Longa, and Benjamin Wesolowski. "New Algorithms for the Deuring Correspondence - Towards Practical and Secure SQISign Signatures". In: *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*. Ed. by Carmit Hazay and Martijn Stam. Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 659–690. DOI: 10.1007/978-3-031-30589-4_23. URL: https://doi.org/10.1007/978-3-031-30589-4_23.

[23] Christina Delfs and Steven D. Galbraith. "Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$". In: *Designs, Codes and Cryptography* 78 (2016), pp. 425–440.

[24] Jonathan Komada Eriksen, Lorenz Panny, Jana Sotáková, and Mattia Veroni. "Deuring for the People: Supersingular Elliptic Curves with Prescribed Endomorphism Ring in General Characteristic". In: *IACR Cryptol. ePrint Arch.* (2023), p. 106. URL: https://eprint.iacr.org/2023/106.

[25] Steven D. Galbraith. *Advances in Elliptic Curve Cryptography, Chapter IX*. Ed. by Ian F. Blake, Gadiel Seroussi, and Nigel P. Smart. Vol. 317. Cambridge University Press, 2005.

[26] Dale Husemöller. *Elliptic Curves, 2nd edition*. Springer, 2004.

[27] David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, Vladimir Soukharev, David Urbanik, Geovandro Pereira, Koray Karabina, and Aaron Hutchinson. *SIKE*. Tech. rep. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions. National Institute of Standards and Technology, 2022.

[28] Don Johnson, Alfred Menezes, and Scott A. Vanstone. "The Elliptic Curve Digital Signature Algorithm (ECDSA)". In: *Int. J. Inf. Sec.* 1.1 (2001), pp. 36–63. DOI: 10.1007/s102070100002. URL: https://doi.org/10.1007/s102070100002.

[29] Simon Josefsson and Ilari Liusvaara. "Edwards-Curve Digital Signature Algorithm (EdDSA)". In: *RFC* 8032 (2017), pp. 1–60. DOI: 10.17487/RFC8032. URL: https://doi.org/10.17487/RFC8032.

[30] David Kohel, Kristin Lauter, Christophe Petit, and Jean-Pierre Tignol. "On the quaternion-isogeny path problem". In: *LMS Journal of Computation and Mathematics* 17.A (2014), pp. 418–432.

[31] Kaizhan Lin, Weize Wang, Zheng Xu, and Chang-An Zhao. *A Faster Software Implementation of SQISign*. Cryptology ePrint Archive, Paper 2023/753. 2023. URL: https://eprint.iacr.org/2023/753.

[32] Michael Meyer and Steffen Reith. "A Faster Way to the CSIDH". In: *Progress in Cryptology - INDOCRYPT 2018 - 19th International Conference on Cryptology in India, New Delhi, India, December 9-12, 2018, Proceedings*. Ed. by Debrup Chakraborty and Tetsu Iwata. Vol. 11356. Lecture Notes in Computer Science.

Springer, 2018, pp. 137–152. DOI: 10.1007/978-3-030-05378-9_8. URL: https://doi.org/10.1007/978-3-030-05378-9_8.

[33] Aurel Page and Benjamin Wesolowski. "The supersingular Endomorphism Ring and One Endomorphism problems are equivalent". In: *CoRR* abs/2309.10432 (2023). DOI: 10.48550/arXiv.2309.10432. arXiv: 2309.10432. URL: https://doi.org/10.48550/arXiv.2309.10432.

[34] Joost Renes and Benjamin Smith. "qDSA: small and secure digital signatures with curve-based Diffie–Hellman key pairs". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2017, pp. 273–302.

[35] Michael Scott. "A note on the calculation of some functions in finite fields: Tricks of the trade". In: *Cryptology ePrint Archive* (2020).

[36] Peter W. Shor. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer". In: *SIAM review* 41.2 (1999), pp. 303–332.

[37] Victor Shoup. "Efficient computation of minimal polynomials in algebraic extensions of finite fields". In: *Proceedings of the 1999 international symposium on Symbolic and algebraic computation*. 1999, pp. 53–58.

[38] Joseph H. Silverman. *The arithmetic of elliptic curves*. Vol. 106. Springer, 2009.

[39] National Institute of Standards and Technology (NIST). *Call for Additional Digital Signature Schemes for the Post-Quantum Cryptography Standardization Process*. 2022. URL: https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf.

[40] Kiminori Tsukazaki. "Explicit isogenies of elliptic curves". PhD thesis. University of Warwick, 2013.

[41] Jacques Vélu. "Isogénies entre courbes elliptiques". In: *Comptes-Rendus de l'Académie des Sciences* 273 (1971), pp. 238–241.

[42] John Voight. *Quaternion algebras*. Springer Nature, 2021.

[43] Benjamin Wesolowski. "The supersingular isogeny path and endomorphism ring problems are equivalent". In: *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2022, pp. 1100–1111.

A Curve arithmetic

In this section we describe in detail the known techniques from literature that allow for general improvements to verification, but that are not included in SQIsign (NIST).

We use $\texttt{xDBL}(x_P)$ to denote x-only point doubling of a point P and $\texttt{xADD}(x_P, x_Q, x_{P-Q})$ to denote x-only differential addition of points P and Q. We use $\texttt{xMUL}(x_P, m)$ to denote x-only scalar multiplication of a point P by the scalar m.

A.1 Faster scalar multiplications

We describe three improvements to the performance of xMUL, that can be applied in different situations during verification.

1. **Affine A.** Throughout verification and specifically in FindBasis and FindKernel, we work with the Montgomery coordinate A in projective form. However, some operations, such as computing the point difference x_{P-Q} given x_P and x_Q require A in affine form. Having an affine A allows an additional speed-up, as xDBL requires one $\mathbb{F}_{p^2}$-multiplication less in this case. Thus, xMUL with affine A is cheaper by 3 $\mathbf{M}$ per bit of the scalar.
2. **Affine points.** Using batched inversion, whenever we require A in affine form we can get x_P and x_Q in affine form for almost no extra cost. An xMUL with affine x_P or x_Q saves another $\mathbb{F}_{p^2}$-multiplication, hence again 3 $\mathbf{M}$ per bit of the scalar.
3. **Small x-coordinate.** For a point P with $x_P = a + bi$ with small a and b, we can replace an $\mathbb{F}_{p^2}$-multiplication by x_P with $a + b$ additions. This, in turn, saves almost 3 $\mathbf{M}$ per bit of the scalar in any xMUL of x_P.

As we can force P and Q to have $b \in \{0, 1\}$ and small a when sampling them in FindBasis, these points are affine and have small x-coordinates. Together with the affine A, this saves almost 9 $\mathbf{M}$ per bit for such scalar multiplications, saving roughly 27% per xMUL. We call such an xMUL a *fast* xMUL. Whenever xMUL uses 2 of these optimisations, we call it *semi-fast*.

Whenever possible, we use differential addition chains [7] to improve scalar multiplications by certain system parameters, such as $\frac{p+1}{2^f}$. In particular, we will only need to multiply by a few, predetermined scalars, and therefore we follow the method described by Cervantes-Vázquez, Chenu, Chi-Domínguez, Feo, Rodríguez-Henríquez, and Smith [11, §4.2]. Our optimal differential addition chains were precomputed using the CTIDH software [5].

A.2 Faster square roots

We apply several techniques from the literature to further optimise low-level arithmetic in all of verification. The most significant of these is implementing faster square roots in $\mathbb{F}_{p^2}$ [35, §5.3], which decreases the cost of finding square roots to two $\mathbb{F}_p$-exponentiations and a few multiplications.

A.3 Projective point difference

The implementation of SQIsign (NIST) switches between affine and projective representations for x_P, x_Q and A within each block. It does so to be able to derive the point difference x_{P-Q} from x_P and x_Q in order to complete the basis P, Q in terms of x-coordinates. However, it is possible to compute the point difference entirely projectively using Proposition 3 from [34]. This allows us to stay projective during the SQIsign (NIST) verification until we reach E_2, where we do normalization of the curve. This saves costly inversions during verification and has the additional benefit of improved elegance for SQIsign (NIST).

However, in our variant of verification, we make no use of projective point difference, as the improvements of Section 5 seem to outperform this already.

B Algorithms

The bottleneck of SQIsign verification is the computation of an isogeny of fixed degree 2^e, which is computed as $\lceil e/f \rceil$ isogenies of degree 2^f, where $f \leq e$. Each such 2^f-isogeny is called a *block*. In this section, we present algorithms for the computation of a single block in verification of SQIsign (NIST) (see Algorithm 2) and the computation using the improvements described in Sections 5 and 6 (see Algorithm 3).

Algorithm 2 Single block in verification of SQIsign (NIST)

```
Input: Affine coeff. $A \in \mathbb{F}_p$, a basis $x_P, x_Q, x_{P-Q}$ for $E_A[2^f]$ with $Q$ above $(0,0)$ and $s \in \mathbb{Z}/2^f\mathbb{Z}$ defining a kernel
Output: Affine coeff. $A' \in \mathbb{F}_p$ as the codomain of $E_A \to E_{A'}$ of degree $2^f$, with a basis $x_P, x_Q, x_{P-Q}$ for $E'_A[2^f]$ with $Q$ above $(0,0)$
$K \leftarrow$ 3ptLadder$(x_P, x_Q, x_{P-Q}, s, A)$
$A_{\text{proj.}}, x_Q \leftarrow$ FourIsogenyChain$(K, x_Q, A)$
$A, x_Q \leftarrow$ ProjectiveToAffine$(A_{\text{proj.}}, x_Q)$
$x_P, x_{P-Q} \leftarrow$ CompleteBasis$_{2^f}(x_Q, A)$
return $A, x_P, x_Q, x_{P-Q}$
```

C Performance of optimised verification

The optimisations for compressed variants from Section 5 and Section 6 allow for several variants of verification, depending on using seeds and pushing Q through isogenies. We summarise the four resulting approaches and measure their performance.

Algorithm 3 Single block in verification using improvements of Section 5

```
Input: Projective coeff. $A \in \mathbb{F}_p$, a seed $(n, m)$ and $s \in \mathbb{Z}/2^f\mathbb{Z}$ defining a kernel
Output: Affine coeff. $A' \in \mathbb{F}_p$ as the codomain of $E_A \to E_{A'}$ of degree $2^f$
$x_P \leftarrow \texttt{SmallNonSquare}(m)$, $x_Q \leftarrow n$
$x_{P-Q} \leftarrow \texttt{PointDifference}(x_P, x_Q, A)$        ▷ implicit basis $x_P$, $x_Q$, $x_{P-Q}$
$K \leftarrow \texttt{3ptLadder}(x_P, x_Q, x_{P-Q}, s, A)$
$K \leftarrow \texttt{xMUL}(x_K, \frac{p+1}{2^f}, A)$        ▷ semi-fast xMUL
$A_{\text{proj.}} \leftarrow \texttt{FourIsogenyChain}(K, A)$
$A \leftarrow \texttt{ProjectiveToAffine}(A_{\text{proj.}})$
return $A$
```

C.1 Four approaches for verification

To obtain our measurements, we combine our optimisations to give four different approaches to perform SQIsign verification, specifically optimised for $f \geq \lambda$. Firstly, we either push Q through φ in every block, or sample Q. Secondly, we either sample the basis or seed it.

Pushing Q, sampling P without seed. This variant is closest to the original SQIsign (NIST) and SQIsign (LWXZ) implementations. It is the optimal version for non-seeded verification for $f \leq 128$, using the general optimisations from Section 5.2 and the challenge optimisations from Section 5.4.

Not pushing Q, sampling both P and Q without seed. This variant competes with the previous version in terms of signature size. Due to Section 5.3, sampling a new Q is more efficient than pushing Q for large f.[26] This is the optimal version for non-seeded verification for $f > 128$, and additionally uses the optimisations from Section 5.3.

Pushing Q, sampling P with seed. This variant only adds seeds to the signature to describe x_P. As such, it lies between the other three variants in terms of both signature size and speed. The signature is 1 byte per block larger than the unseeded variants, and 1 byte per block smaller than the variant where x_Q is seeded too. In terms of speed, it is faster than the variants where P is unseeded, but slower than the variant where Q is seeded too. It uses the optimisations from Sections 5.2 and 5.4, but cannot benefit from the kernel computation via implicit bases from Section 5.3.

[26] Based on benchmarking results, we sample Q with $x = n\alpha$ for $f < 200$ and directly for $f \geq 200$.

Not pushing Q and sampling both P and Q with seed. This is the fastest compressed version that we present in this work. Although it adds 2 bytes per block, the small number of blocks n for large f makes the total increase in signature size small. All the optimisations from Section 5 now apply: we additionally have fast `xMUL` for Q, as well as the optimised implicit basis method to compute the kernel and optimised challenge. An algorithmic description of a single block in this version is given in Algorithm 3.

C.2 Performance benchmark

We benchmarked these four approaches according to our cost metric by taking the average over 1024 random signatures. The results are given in Figure 4 showing the significant increase in performance compared to SQIsign (NIST) and SQIsign (LWXZ), as well as the additional performance gained from seeding. For comparison, we also show the performance when using uncompressed signatures, serving as a lower bound for the cost.

D Detailed information on primes

We give more details on the specific primes used in Section 7.

D.1 Details on p_7

The prime p_7 is used for a verification with $n = 7$ blocks. It achieves $f = 145$, with T given as below.

$$\begin{aligned}
p_7 &= \texttt{0x309c04bcaedbb0134cca8373e439ffffffffffffffffffffffffffffffffffff}\\
T &= 3^7 \cdot 5^4 \cdot 7^2 \cdot 11 \cdot 13 \cdot 17 \cdot 19^2 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 \cdot 53^2 \cdot 59^3 \cdot 61 \cdot 67\\
&\quad \cdot 71 \cdot 73 \cdot 79 \cdot 109 \cdot 113 \cdot 131 \cdot 157 \cdot 181 \cdot 193 \cdot 223 \cdot 239 \cdot 241 \cdot 271 \cdot 283 \cdot 311^3\\
&\quad \cdot 317^3 \cdot 331 \cdot 349 \cdot 503^2 \cdot 859 \cdot 997\\
\textsc{SigningCost}_{p_7}(T) &= 4137.91235
\end{aligned}$$

The field of definition for the various torsion groups we work with can be found in Table 3.

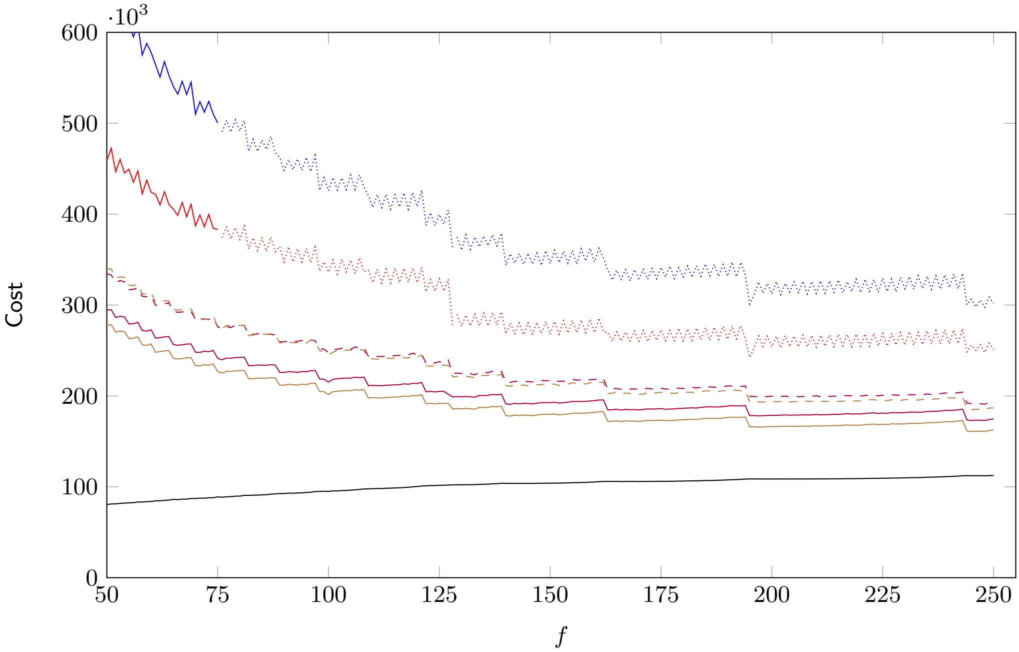


Fig. 4: Extended version of Figure 3 showing the cost in $\mathbb{F}_p$-multiplications for verification at NIST-I security level, for varying f and $p^{(f)}$, averaged over 1024 runs per prime. In addition to SQIsign (NIST) in blue, and SQIsign (LWXZ) in red, it shows all ApresSQI variants: In purple is the performance of ApresSQI when pushing Q, with dashed blue when not seeding P. In brown is the performance of ApresSQI when not pushing Q, with dashed brown when not seeding P, Q. The performance of uncompressed ApresSQI is shown in black.

D.2 Details on p_4

The prime p_4 is used for a verification with $n = 4$ blocks. It achieves $f = 246$, with T given as below.

$$
\begin{aligned}
p_4 &= \texttt{0x323ffffffffffffffffffffffffffffffffffffffffffffffffffffffffff} \\
T &= 3^3 \cdot 5^2 \cdot 7^2 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 \cdot 53 \cdot 59 \cdot 61 \cdot 67 \cdot 71 \\
&\quad \cdot 73 \cdot 79 \cdot 83 \cdot 89 \cdot 97 \cdot 101 \cdot 103 \cdot 107 \cdot 109 \cdot 113 \cdot 127 \cdot 149 \cdot 151 \cdot 157 \cdot 163 \cdot 181 \\
&\quad \cdot 197 \cdot 211 \cdot 229 \cdot 241 \cdot 271 \cdot 317 \cdot 397 \cdot 577 \cdot 593 \cdot 641 \cdot 661 \cdot 757 \cdot 1069 \cdot 2293 \\
\textsc{SigningCost}_{p_4}(T) &= 9632.7307
\end{aligned}
$$

The field of definition for the various torsion groups can be found in Table 4.

Table 3: Torsion groups $E[N]$ and their minimal field $E(\mathbb{F}_{p^{2k}})$ for the prime p_7

k	N
1	$3^7, 53^2, 59^3, 61, 79, 283, 311^3, 317^3, 349, 503^2, 859, 997$
3	$13, 109, 223, 331$
4	17
5	$11, 31, 71, 241, 271$
6	157
7	$7^2, 29, 43, 239$
8	113
9	19^2
10	$5^4, 41$
11	$23, 67$
12	193
13	131
15	181
18	$37, 73$
23	47

Table 4: Torsion groups $E[N]$ and their minimal field $E(\mathbb{F}_{p^{2k}})$ for the prime p_4

k	N
1	$67, 73, 757$
2	$317, 2293$
3	$37, 127, 1069$
4	593
5	$11, 31, 71, 661$
6	13
7	43
8	$17, 113$
9	$3^3, 19, 181, 577$
10	$5^2, 61, 641$
11	$23, 89$
14	$29, 197$
18	397
19	229
20	41
21	7^2
23	47
25	151
26	53
27	$109, 163, 271$
29	59
30	241
35	211
37	149
39	$79, 157$
41	83
48	97
50	101
51	103
53	107

Part 3

Oriented Endomorphism Rings

To all of the people in T-town:
Thanks for not letting me down.
Eight hell of a years,
ended in tears.
This thesis was written by Clown.

Hanne Æsoy Nes,
Jonathan Komada Eriksen,
May 2021 (upd. May 2024)

Computing Orientations from the Endomorphism Ring of Supersingular Curves and Applications

Jonathan Komada Eriksen and Antonin Leroux

In submission. Full version available at: eprint.iacr.org/2024/146.pdf

Computing Orientations from the Endomorphism Ring of Supersingular Curves and Applications

Jonathan Komada Eriksen[3], Antonin Leroux[1,2]

[1] DGA-MI, Bruz, France antonin.leroux@polytechnique.org
[2] IRMAR, Université de Rennes, France
[3] Norwegian University of Science and Technology, Norway

Abstract. This work introduces several algorithms related to the computation of orientations in endomorphism rings of supersingular elliptic curves. This problem boils down to representing integers by ternary quadratic forms, and it is at the heart of several results regarding the security of oriented-curves in isogeny-based cryptography.

Our main contribution is to show that there exists efficient algorithms that can solve this problem for quadratic orders of discriminant n up to $O(p^{4/3})$. Our approach improves upon previous results by increasing this bound from $O(p)$ to $O(p^{4/3})$ and removing some heuristics.

We introduce several variants of our new algorithm and provide a careful analysis of their asymptotic running time (without heuristic when it is possible). The best proven asymptotic complexity of one of our variant is $O(n^{3/4}/p)$ in average. The best heuristic variant has a complexity of $O(p^{1/3})$ for big enough n.

We then introduce several results regarding the computation of ideals between oriented orders. The first application of this is a simplification of the known reduction from vectorization to computing the endomorphism ring, removing the assumption on the factorization of the discriminant. As a second application, we relate the problem of computing fixed-degree isogenies between supersingular curves to the problem of computing orientations in endomorphism rings, and we show that for a large range of degree d, our new algorithms improve on the state-of-the-art, and in important special cases, the range of degree d for which there exist a polynomial-time algorithm is increased. In the most special case we consider, when both curves are oriented by a small degree endomorphism, we show heuristically that our techniques allow the computation of isogenies of any degree, assuming they exist.

1 Introduction

Isogeny-based cryptography uses supersingular elliptic curves and isogenies between them to construct cryptographic schemes. An essential part of isogeny-based cryptography is the Deuring correspondence, relating supersingular elliptic curves over $\overline{\mathbb{F}}_p$ to maximal orders in a quaternion algebra ramified at p and ∞, and isogenies to ideals, by passing to the endomorphism ring of the curve.

One particular flavour of isogeny-based schemes [3, 5, 9] use the extra information of an *orientation*, which is an embedding of a quadratic imaginary order inside the endomorphism ring. This subring corresponds to an embedding of an imaginary quadratic order $\mathfrak{O}$ into the endomorphism ring, which in turn allows one to consider the action of $\mathfrak{O}$-ideals on the curves (primitively) oriented by $\mathfrak{O}$ through $\mathfrak{O}$-oriented isogenies. It is a well known fact that $\mathrm{Cl}(\mathfrak{O})$ acts freely on the set of primitively $\mathfrak{O}$-oriented curves (up to oriented isomorphisms) in one or two orbits [15].

An important part of the study of the schemes using these oriented curves and isogenies it to understand the link of oriented problems with generic non-oriented problems. One of the main object of study in this context is the embedding problem which was first studied in [19] (although not under that name). We present it as Problem 1.

Problem 1. **(Quaternion order embedding problem.)** Let p be a prime number, let $\mathcal{O}$ be a maximal order inside $B_{p,\infty}$ and let t, n be such that there exists an element of norm n and trace t in $\mathcal{O}$. Find $\alpha \in \mathcal{O}$ with

$$n(\alpha) = n, \ \ \mathrm{tr}(\alpha) = t \tag{1}$$

Related Works. Oriented curves first appeared in isogeny-based cryptography with the CSIDH group action [3]. However, they were not defined as such at the time. The notion of orientation was introduced formally by Kohel and Colo in [5] together wiht a new group action called OSIDH. Some of the results of [5] were refined by Onuki [15]. These works introduced generic hard problems such as $\mathfrak{O}$-vectorization.

At first, the only applications of orientations were related to these group actions, but a broader link with the other areas of isogeny-based cryptography was demonstrated by De Feo et al. in [7] with the introduction of the $\mathfrak{O}$-uber isogeny problem. The authors of [7] provided in particular some reductions between flavours of the $\mathfrak{O}$-uber isogeny problem and generic isogeny computation problems.

In 2022, Wesolowski provided a much more complete picture in [19] by studying all orientation-related problems and providing several reductions between them, and generic problems such as the endomorphism ring problem. In particular, Wesolowski proposed the first algorithm to solve the quaternion order embedding problem when the discriminant is smaller than $\sqrt{p}$, and proved some relations between the $\mathfrak{O}$-vectorization, the $\mathfrak{O}$-uber isogeny, and problems related to the computation of endomorphism rings (with or without the knowledge of an orientation).

An improved heuristic algorithm to solve the embedding problem was introduced in [1] that increases the bound for when the embedding problem is solvable in polynomial time from disc $\mathfrak{O} = O(\sqrt{p})$ to disc $\mathfrak{O} = O(p)$.

In [12], Leroux proved a lower bound on the number of oriented curves by using quaternion orders generated by two non-commuting quadratic orders. The same ideas are going to be crucial in our new algorithms.

An algorithm to solve the embedding problem can be used to find fixed degree isogeny between supersingular elliptic curves. This is an important problem in isogeny-based cryptography that was first studied from the quaternionic perspective in [11] with the famous KLPT algorithm. This algorithm has found numerous applications in cryptography in the study of the Deuring correspondence (see the reductions of [8] or the signature scheme from [6] for instance). Understanding and improving the known algorithms to find isogenies of fixed degree between supersingular curves is an important task. While previous literature had been focusing on identifying cases for which there was an polynomial-time algorithm (such as KLPT), the recent article [2] was the first one to provide a generic analysis of the run-time of such algorithms in ranges of input where the running-time is not known to be polynomial.

1.1 Our Contributions

In this work, we study orientations purely on the quaternion side. Our main contribution is a set of new algorithms for solving the quaternion order embedding problem (Problem 1), which can be executed in polynomial time for disc $\mathfrak{O}$ up to $O(p^{4/3})$.

GenericOrderEmbedding, our first algorithm, treats the generic case of an arbitrary quaternion order. It's complexity depends on the size of the first, second, and third successive minima of the ternary quadratic form associated to $\mathcal{O}$. When $\mathcal{O}$ is a random quaternion order of discriminant Δ the expected running time is polynomial when disc $\mathfrak{O} = O(\Delta^{4/3})$.

From there, we deduce two other algorithms. MaximalOrderEmbeddingEichler uses GenericOrderEmbedding as a building block by applying it on several Eichler sub-orders of the maximal provided in input. We show that the average running time is asymptotically better than a direct application of GenericOrderEmbedding.

In some good cases where $\mathcal{O}$ contains a particularly small element, we can go beyound the $O(p^{4/3})$ bound at the cost of using a factorization oracle, under some heuristics. The resulting algorithm GenericOrderEmbeddingFactorization can be seen as a generalization of the algorithm from [1], and in the best cases where the $\mathcal{O}$ contains an element of norm $O(1)$, it runs in polynomial time for any discriminant. Further, for any order, the runtime is always upper bounded as $O(p^{1/3})$, independent of the size of the discriminant.

In the second part, we study ideals between oriented quaternion orders. We show that when the orientation of the quaternion orders induce the same orientation of K into $B_{p,\infty}$, their connecting ideal is always generated by the image of a quadratic ideal. We apply this result to give a new, simple reduction to show that the $\mathfrak{O}$-vectorization problem reduces to the endomorphism ring problem, a result previously only known for when the factorization of disc $\mathfrak{O}$ was known, and assuming $\mathfrak{O}$ has a small number of genera [19, Theorem 2].

We also give a heuristic reduction from the problem of computing equivalent ideals of a given norm to the quaternion order embedding problem, and show that in important special cases, our algorithms improves the range for which this is solvable in polynomial time. In particular, in the special case where the two maximal orders are optimally embedded by quadratic orders of very small discriminant, it is possible to find equivalent ideals of any norm efficiently. We also obtain a heuristic improvement in the best known asymptotic complexity to solve this problem in the generic case, showing that it is always solvable in time $O(p^{2/3})$, improving on the results from [2] for a wide variety of degrees d.

We implement our algorithms in SageMath [17]. The implementation can be found at:

`https://github.com/Jonathke/Computing-Optimal-Embeddings`

1.2 Technical Overview

Let us take an order $\mathcal{O}$ of dscriminant Δ, and elements t, n, α as in Problem 1.

Overview of the algorithms. Our new algorithms to find elements of given norm and trace are mainly built upon an oracle to find trace pairings, i.e.the value of the trace of the product of the element α with some elements β of $\mathcal{O}$. This oracle is built by looking at the discriminant of the quaternion order $\mathbb{Z}[1, \alpha, \beta, \alpha\beta]$ and seeing that its discriminant must be divided by Δ when α and β do not commute. This gives an equation on $\mathrm{tr}(\alpha\beta)$ modulo Δ. And this equation is enough to recover the value over $\mathbb{Z}$ when $n(\alpha\beta) < \Delta^2$.

We obtain our algorithm GenericOrderEmbedding by applying this idea on a reduced basis $1, \beta_1, \beta_2, \beta_3$ of $\mathcal{O}$ and enumerating all possible solutions until the correct one is found. As for a random order $\mathcal{O}$ we can expect $n(\beta_1) \approx n(\beta_2) \approx n(\beta_3) \approx \Delta^{2/3}$, this will be efficient to recover α when $n = O(\Delta^{4/3})$ and we can show that asymptotically (when n grows and p remains fixed) the complexity of this algorithm is $O(n^{3/2}/p^2)$.

Our algorithm MaximalOrderEmbeddingEichler is obtained by trying to apply GenericOrderEmbedding on all Eichler sub-order of order N (where N is chosen to ensure that each execution GenericOrderEmbedding should be polynomial in average and that there is one execution that will succeed). We show that the average running time of this algorithm is $O(n^{3/4}/p)$.

Finally, in cases where $n(\beta_1)$ is smaller than the expected $\Delta^{2/3}$, the trace pairing $\mathrm{tr}(\alpha\beta_1)$ will be much smaller than $\mathrm{tr}(\alpha\beta_j)$ for $1 < j \leq 3$. Thus, it will be possible to determine $\mathrm{tr}(\alpha\beta_1)$ exactly for values of n bigger than $\Delta^{4/3}$. In those cases, we can exploit the knowledge of $\mathrm{tr}(\alpha\beta_1)$ to translate the embedding problem to a problem of representing some integer by some binary quadratic form. It is well known that such equation can be solved in polynomial time with the help of a factorization oracle. This yields the GenericOrderEmbeddingFactorization algorithm.

Acknowledgement

The authors would like to thank Benjamin Wesolowski for a fruitful discussion that motivated this work. We would also like to thank Wouter Castryck, Gioella Lorenzon, Riccardo

Invernizzi and Frederik Vercauteren, with whom useful discussions among other things lead to a cleaner description of GenericOrderEmbeddingFactorization, the introduction of Heuristic 2, and subsequent heuristic running times. Finally, we thank the members of Group project F at the 2023 Bristol isogeny workshop for useful discussions about several aspects of this work. We also thank the organizers of this workshop for providing a nice framework for these discussions.

2 Mathematical Background

A *quaternion algebra* B is a four dimensional $\mathbb{Q}$-algebra with a $\mathbb{Q}$-basis $1, i, j$, satisfying

$$i^2 = a, j^2 = b, k = ij = -ji,$$

for some $a, b \in \mathbb{Q}^\times$. Elements $\alpha = x+iy+jz+kw \in B$ have a conjugate $\bar{\alpha} = x-iy-jz-kw$, and from this we define the reduced norm $\mathrm{n}(\alpha) := \alpha\bar{\alpha}$ and reduced trace $\mathrm{tr}(\alpha) := \alpha + \bar{\alpha}$.

The values a, b determine the places where B *ramify*, which again determines B up to isomorphism. In this work, we fix a prime p, and focus on the quaternion algebra $B_{p,\infty}$ ramified at p and ∞.

A *lattice* in $B_{p,\infty}$ is a $\mathbb{Z}$-submodule $L \subseteq B_{p,\infty}$ of rank 4. Lattices have an invariant called the discriminant, defined as

$$\mathrm{disc}\ L = \det\left(\mathrm{tr}(\beta_i\beta_j)_{i,j}\right)$$

where $\beta_1, \beta_2, \beta_3, \beta_4$ is a $\mathbb{Z}$-basis of L. A lattice $\mathcal{O}$ is called an *order* if it is also a subring of $B_{p,\infty}$, i.e. it contains 1, and is closed under multiplication. The discriminant of an order is always a square, hence we can define the *reduced discriminant*

$$\mathrm{discrd}\ \mathcal{O} = \sqrt{\mathrm{disc}\ \mathcal{O}} \in \mathbb{Z}$$

In $B_{p,\infty}$, orders $\mathcal{O}$ always satisfy

$$\mathrm{discrd}\ \mathcal{O} = pN,$$

where $N := [\mathcal{O}_0 : \mathcal{O}]$, for some maximal order $\mathcal{O}_0$ containing $\mathcal{O}$.

In the rest of this document, we will always use the reduced discriminant of quaternion order despite very often using the word *discriminant* and using the notation $\mathrm{disc}\ \mathcal{O}$.

2.1 On Successive Minimas in a Quaternion Order

We define the successive minimas of a quaternion order $\mathcal{O}$ to be the successive minimas of $\mathcal{O}/\mathbb{Z}$.

Below, we prove several simple results bounding the successive minimas of quaternion orders. Most of those results are folklore and/or very easy to prove but we restate them for convenience.

In all this section, $\mathcal{O}$ is a quaternion order of reduced discriminant Δ and $\beta_1, \beta_2, \beta_3$ realizes the successive minimas of $\mathcal{O}$.

Proposition 1. *(Minkowski)* $\frac{4}{3}\Delta^2 \leq n(\beta_1)n(\beta_2)n(\beta_3) \leq 8\Delta^2$

Lemma 1. $n(\beta_1) \leq 2\Delta^{2/3}$.

Proof. This follows from combining Proposition 1 with $n(\beta_2)n(\beta_3) \geq n(\beta_1)^2$. □

Lemma 2. $n(\beta_2) \leq 2\sqrt{2}\Delta/\sqrt{n(\beta_1)}$.

Proof. By Proposition 1, $n(\beta_1)n(\beta_2)^2 \leq n(\beta_1)n(\beta_2)n(\beta_3) \leq 8\Delta^2$, and this proves the result. □

Lemma 3. $n(\beta_2) \geq \Delta/(4n(\beta_1))$.

Proof. The quaternion order $\mathbb{Z}[1, \beta_1, \beta_2, \beta_1\ \beta_2]$ is contained in $\mathcal{O}$. Thus, by Proposition 2, we have $\Delta \leq 4n(\beta_1)n(\beta_2)$. □

Lemma 4. $n(\beta_1) \leq \frac{2\sqrt{2}\Delta}{\sqrt{n(\beta_3)}}$

Proof. Combining Proposition 1 with $n(\beta_1)^2 n(\beta_3) \leq n(\beta_1)n(\beta_2)n(\beta_3) \leq 8\Delta^2$ proves the result. □

Lemma 5. $n(\beta_3) \leq 32\Delta$

Proof. The result follows from the combination of Lemma 3 with Proposition 1. □

2.2 Oriented Orders

The main focus in this paper is on *optimal embeddings*. Our main motivation is the relation to *primitively* $\mathfrak{O}$*-oriented curves*, defined as the pair (E, ι), where E is a supersingular elliptic curve, and $\iota : K \hookrightarrow \mathrm{End}(E) \otimes \mathbb{Q}$ is an optimal embedding of $\mathfrak{O}$ into $\mathrm{End}(E)$, i.e. such that

$$\iota_{|\mathfrak{O}} : \mathfrak{O} \hookrightarrow \mathrm{End}(E)$$

satisfies

$$\iota(K) \cap \mathcal{O} = \iota(\mathfrak{O}).$$

Hence, we introduce the analogous notation, which will be used repeatedly in Section 4:

Definition 1. *Let K an imaginary quadratic field, with $\mathfrak{O} \subseteq K$ an imaginary quadratic order and let B be a definite quaternion algebra over $\mathbb{Q}$ with $\mathcal{O} \subseteq B$ an order. Given an embedding $\iota : K \hookrightarrow B$, we can define a $\mathfrak{O}$-oriented order to be the pair $(\mathcal{O}, \iota)$, whenever $\iota(\mathfrak{O}) \subseteq \mathcal{O}$. Further, $(\mathcal{O}, \iota)$ is said to be a primitively $\mathfrak{O}$-oriented order if $\iota(\mathfrak{O}) = \mathcal{O}$.*

2.3 On the Order Generated by two Quaternion Elements

Give two integral elements $\alpha_1, \alpha_2 \in B$ that does not commute, $\mathbb{Z}\langle\alpha_1, \alpha_2\rangle \subseteq B$ is an order, with discriminant given by the following proposition:

Proposition 2. *[10, Chapter 7] Let $\mathfrak{O}_i$ be quadratic orders equal to $\mathbb{Z}[\alpha_i]$ for $i = 1, 2$ such that α_1, α_2 are not commuting. Let $D_i = \operatorname{disc} \mathfrak{O}_i$, $t_i = \operatorname{tr}(\alpha_i)$ for $i \in \{1, 2\}$ and $s = \operatorname{tr}(\alpha_1\alpha_2)$, then*

$$\operatorname{disc} \mathbb{Z}\langle\alpha_1, \alpha_2\rangle = (D_1D_2 - (t_1t_2 - 2s)^2)/4$$

3 Algorithms to Solve the Quaternion Embedding Problem

In this section, we present several algorithms to solve the quaternion embedding problem.

3.1 A First Algorithm for a Generic Order.

Our first algorithm GenericOrderEmbedding makes use of the formula stated in Proposition 2 on the discriminant of the quaternion order generated by two elements to produce a trace pairing oracle modulo the discriminant of the order $\mathcal{O}$.

Algorithm 1 GenericOrderEmbedding($\mathcal{O}, t, n$)

```
Input: A quaternion order $\mathcal{O} \subset B_{p,\infty}$ of discriminant $\Delta$, two integers $t, n \in \mathbb{Z}$ such that there
    exists an element of trace $t$ and norm $n$ in $\mathcal{O}$.
Output: $\perp$ or $\alpha \in \mathcal{O}$ with $n(\alpha) = n$ and $\operatorname{tr}(\alpha) = t$.
Compute a Minkowski reduced basis $1, \beta_1, \beta_2, \beta_3$ of $\mathcal{O}$.
Compute $D_i = \operatorname{tr}(\beta_i)^2 - 4n(\beta_i)$ for $1 \leq i \leq 3$, and $D = t^2 - 4n$.
Compute $s_i$ a square root of $DD_i \mod \Delta$.
for $t_1, t_2, t_3 \in [-\sqrt{4nn(\beta_1)}, \sqrt{4nn(\beta_1)}] \times [-\sqrt{4nn(\beta_2)}, \sqrt{4nn(\beta_2)}] \times [-\sqrt{4nn(\beta_3)}, \sqrt{4nn(\beta_3)}]$
    such that $t_i = (1/2)(\pm s_i + t\operatorname{tr}(\beta_i)) \mod \Delta$ do
    Compute $\alpha$ the element such that $\operatorname{tr}(\alpha) = t$, and $\operatorname{tr}(\alpha\beta_i) = t_i$ for $1 \leq i \leq 3$.
    if $n(\alpha) = n$ then
        Return $\alpha$.
    end if
end for
return Return $\perp$.
```

Proposition 3. *Let $\mathcal{O} \subset B_{p,\infty}$ be a quaternion order of discriminant Δ (whose factorization is known) and Minkowski reduced basis $1, \beta_1, \beta_2, \beta_3$. Let t, n be two integers such* *that there exists an element of norm n and trace t in $\mathcal{O}$, the* GenericOrderEmbedding *will output an element α in $\mathcal{O}$ with the correct trace and norm and runs in*

$$O\left(\left\lceil 8\frac{\sqrt{nn(\beta_1)}}{\Delta}\right\rceil \left\lceil 8\frac{\sqrt{nn(\beta_2)}}{\Delta}\right\rceil \left\lceil 8\frac{\sqrt{nn(\beta_3)}}{\Delta}\right\rceil \mathrm{polylog}(\Delta n)\right)$$

Proof. Since $1, \beta_1, \beta_2, \beta_3$ is a basis of $\mathcal{O}$, any element $\alpha \in \mathcal{O}$ is uniquely determined by the values $\mathrm{tr}(\alpha)$ and $\mathrm{tr}(\alpha\beta_i)$ for $1 \leq i \leq 3$.

For any element α of $\mathcal{O}$, the quaternion order $\mathbb{Z}\langle \alpha, \beta_i \rangle$ is contained in $\mathcal{O}$ and so its discriminant is divisible by Δ. Thus, with the formula given in Proposition 2, we get that if α has trace t and norm n, we must have $DD_i = (t\mathrm{tr}(\beta_i) - 2\mathrm{tr}(\alpha\beta_i))^2 \mod \Delta$ which gives $\mathrm{tr}(\alpha\beta_i) = \pm s_i + t\mathrm{tr}(\beta_i) \mod \Delta$ where $s_i^2 = DD_i \mod \Delta$.

Moreover, since every quadratic order in the quaternion order $\mathcal{O}$ has negative discriminant we must have $\mathrm{tr}(\alpha\beta_i)^2 \leq 4nn(\beta_i)$.

Thus, assuming that there exists an element of norm n and trace t in $\mathcal{O}$, then there will be one triple of value t_1, t_2, t_3 that will lead to a corect element α.

For each $1 \leq i \leq 3$, there are less than $\left\lceil 8\frac{\sqrt{nn(\beta_i)}}{\Delta}\right\rceil$ values of t_i that satisfy the constraint $\mod \Delta$ that are within the desired interval. When the factorization of Δ is known, it is possible to compute the square-root s_i in $O(\mathrm{polylog}(\Delta))$ and all the operations to execute for each triple t_1, t_2, t_3 can be performed in $O(\mathrm{polylog}(\Delta n))$. This proves the result.

Proposition 3 has three interesting corollaries. The first corollary states an asymptotic complexity of GenericOrderEmbedding when n is big compared to Δ.

Corollary 1. *Let $\mathcal{O}, t, n$ be as in Proposition 3, and assume that $n > \Delta^2/64$. Then, the complexity of* GenericOrderEmbedding *is $O\left((n^{3/2}/\Delta^2)\mathrm{polylog}(\Delta n)\right)$.*

Proof. When $n > \Delta^2/64$, we have that $8\sqrt{nn(\beta_i)} > \Delta$ for all $1 \leq i \leq 3$, and so the asymptotics $\lceil 8\sqrt{nn(\beta_i)}/\Delta\rceil = \Theta(\sqrt{nn(\beta_i)}/\Delta)$ holds for any $1 \leq i \leq 3$. Then, we deduce the complexity of GenericOrderEmbedding from Proposition 1.

This second corollary identifies the situation where GenericOrderEmbedding will always be polynomial-time.

Corollary 2. *Let $\mathcal{O}, t, n$ be as in Proposition 3. If $n = O(\Delta)$, then the complexity of* GenericOrderEmbedding *is $O(\mathrm{polylog}(\Delta n))$.*

Proof. By Lemma 5, when $n = O(\Delta)$, $\sqrt{nn(\beta_i)}/\Delta = O(1)$ and the result follows from Proposition 3.

A direct application on maximal orders. We can obtain a first algorithm to solve Problem 1 on input $\mathcal{O}$, by applying directly GenericOrderEmbedding on the maximal order $\mathcal{O}$. In that case, $\Delta = p$, and Corollary 2 proves that our algorithm will be polynomial time when $n = O(p)$. This is already an improvement over the result stated in [1] as it does not rely on any heuristic, but we expect GenericOrderEmbedding to be better than that in average.

In Corollary 3, we give a statement to quantify the number of maximal orders for which the running time of GenericOrderEmbedding is polynomial in term of the fraction $p^{4/3}/n$. The proof of Corollary 3 uses a bound on the number of maximal orders having a non-trivial endomorphism smaller than a given value m that we introduce below as Lemma 6. This result was proven in [14].

Lemma 6. *For any $0 < M < p^{2/3}$, the number of maximal order in $B_{p,\infty}$ containing an element not in $\mathbb{Z}$ of norm smaller than M is $O(M^{3/2})$.*

Corollary 3. *Let $\mathcal{O} \subset B_{p,\infty}$ be a maximal order. Let t, n be as in Proposition 3. Assume further that the order $\mathcal{O}$ is uniformly random among the set of maximal order types.*

There exists a polynomial $P(X) \in \mathbb{Q}[X]$, and constants C_1, C_2 such that for every $\varepsilon > 0$, if $n < C_1 p^{4/3-\varepsilon}$, then the running time of GenericOrderEmbedding *on input $\mathcal{O}, t, n$ is smaller than $P(\log(p))$ with probability bigger than $1 - C_2 p^{-3\varepsilon}$.*

Proof. A maximal order in $B_{p,\infty}$ has discriminant $\Delta = p$.

If $n(\beta_3)n < p^2$, then Proposition 3 implies that the running time of GenericOrderEmbedding is poly-logarithmic in n, p and since $n = O < C_1 p^{4/3-\varepsilon}$ there exists a polynomial $P(X)$ such that the running time is smaller than $P(\log(p))$.

Thus, to prove the result, it suffices to prove that the probability of $n(\beta_3)n$ being bigger than p^2 is smaller than $C_2 p^{-3\varepsilon}$ for some constant C_2.

Using Lemma 4, we can show that if $n(\beta_3)n \geq p^2$, then we must have $n(\beta_1) \leq C\sqrt{n} \leq CC_1 p^{2/3-2\varepsilon}$ for some $C > 0$. By Lemma 6, we know there exists $C' > 0$ such that there are at most $C' p^{1-3\varepsilon}$ maximal orders admitting a non-trivial β_1 of norm smaller than $CC_1 p^{2/3-\varepsilon}$. Since there are $O(p)$ distinct isomorphism classes of maximal orders, we conclude that the probability of finding such a bad maximal orders at random is smaller than $C_2 p^{-3\varepsilon}$ for some constant C_2 and this concludes the proof.

In Appendix A, we outline a heuristic variant of this algorithm, which works with any basis, followed by enumerating close vectors.

3.2 A Better Asymptotic Algorithm to Solve the Embedding Problem.

To solve the embedding problem, we are not restricted to the obvious solution of applying GenericOrderEmbedding on the maximal order given in input.

The goal of this section is to introduce another algorithm MaximalOrderEmbeddingEichler that applies GenericOrderEmbedding on Eichler orders. We will show that the average asymptotic complexity of this algorithm is the square-root of the asymptotic complexity of GenericOrderEmbedding. Unfortunately, despite that improvement, MaximalOrderEmbeddingEichler does not improve on the range of values of n for which the running time is polynomial.

The principle of MaximalOrderEmbeddingEichler is the following: by taking a split prime N in $\mathfrak{O}$, we can deduce that the element α we are looking for must be contained inside an Eichler order of level N contained in $\mathcal{O}$. Thus, we can compute the list of these orders and try to apply GenericOrderEmbedding on all of them until one works. We formalize this idea below as MaximalOrderEmbeddingEichler. The value of N is chosen to ensure that the expected running time of GenericOrderEmbedding on Eichler orders of level N (whose discriminant is pN) is polynomial in $\log(pN)$. This is why we take $N = O(n^{3/4}/p)$. In that case, we can expect the complexity of the algorithm to be $O(N) = O(n^{3/4}/p)$.

Algorithm 2 MaximalOrderEmbeddingEichler($\mathcal{O}, t, n$)

Input: A maximal order $\mathcal{O} \subset B_{p,\infty}$, two integers $t, n \in \mathbb{Z}$ such that there exists an element of trace t and norm n in $\mathcal{O}$.

Output: $\perp$ or $\alpha \in \mathcal{O}$ with $n(\alpha) = n$ and $\mathrm{tr}(\alpha) = t$.

```
Set $D = t^2 - 4n$ and $\mathfrak{O}$ as the maximal order in $\mathbb{Q}(\sqrt{-D})$.
Select a prime $N$ split in $\mathfrak{O}$ such that $N/2 < n^{3/4}/p < N$.
Compute $\mathcal{O}_1, \dots, \mathcal{O}_{N+1}$ the $N+1$ Eichler orders of level $N$ contained in $\mathcal{O}$.
for $i = 1$ to $M$ do
    Compute $\alpha =$ GenericOrderEmbedding($O_i, t, n$).
    if $\alpha \neq \perp$ then
        Return $\alpha$.
    end if
end for
return $\perp$.
```

Despite the informal reasoning outlined above, it is not easy to prove formally what is the complexity of MaximalOrderEmbeddingEichler because there are Eichler orders of level N in $\mathcal{O}$ that will have elements of norm smaller than expected. Nonetheless, we obtain a bound on the average running time by proving that executing MaximalOrderEmbeddingEichler on all maximal orders can be done in $O(n^{3/4+\varepsilon})$ for any $\varepsilon > 0$. This is stated in Proposition 4.

For the proof, we will need another corollary of Proposition 3 to bound the running time of GenericOrderEmbedding in terms of the norm of its successive minimas.

Corollary 4. *Let $\mathcal{O}, t, n$ be as in Proposition 3, with $\beta_1, \beta_2, \beta_3$ the three sucessive minimas of $\mathcal{O}$.*

(i) When $n(\beta_2) < \Delta^2/64n$, the running time of GenericOrderEmbedding *is* $O\left(\left\lceil 16\sqrt{2}\sqrt{\frac{n}{n(\beta_1)n(\beta_2)}}\right\rceil\right)$.

(ii) When $n(\beta_1) < \Delta^2/64n, n(\beta_2) \geq \Delta^2/64n$, the running time of GenericOrderEmbedding *is* $O\left(\frac{n}{\Delta\sqrt{n(\beta_1)}}\right)$.

Proof. For (i), when $n(\beta_2) \leq \Delta^2/64n$, then the first two factors in the complexity given in Proposition 3 are 1, and so the complexity is given by the last term.

From Proposition 1, we get $n(\beta_3) \leq 8\Delta^2/n(\beta_1)n(\beta_2)$ from which we derive

$$\lceil 8\sqrt{nn(\beta_3)}/\Delta\rceil = O(\lceil 16\sqrt{2}\sqrt{\frac{n}{n(\beta_1)n(\beta_2)}}\rceil)$$

For (ii), from $n(\beta_1) < \Delta^2/64n$, we get that the first factor in the complexity stated in Proposition 3 is 1. From $n(\beta_3) \geq n(\beta_2) \geq \Delta^2/64n$, we get that the complexity is

$$O\left(\frac{n\sqrt{n(\beta_2)n(\beta_3)}}{\Delta^2}\right)$$

which can be simplified to

$$O\left(\frac{n}{\Delta\sqrt{n(\beta_1)}}\right)$$

by applying $n(\beta_2)n(\beta_3) = O(\Delta^2/n(\beta_1))$ that we derive from Proposition 1. □

We will also need a lemma to upper-bound the number of Eichler orders admitting an embedding of two non-commuting quadratic orders of discriminant δ_1, δ_2.

Lemma 7. *Let $\mathcal{O}$ be an Eichler order of level N in $B_{p,\infty}$. Let $\mathfrak{O}_1, \mathfrak{O}_2$ be two quadratic imaginary orders of discriminant δ_1, δ_2 (and conductors $f(\delta_1), f(\delta_2)$) such that the $\mathfrak{O}_i$ are optimally embedded inside $\mathcal{O}$ and their embedding is non-commuting.*

Let us take α_1 and α_2 two elements of $\mathcal{O}$ such that optimal embedding of $\mathfrak{O}_i$ is equal to $\mathbb{Z}[\alpha_i]$. Let $\mathcal{O}_{1,2}$ be the quaternion sub-order of $\mathcal{O}$ generated by α_1, α_2 and let $s = \mathrm{tr}(\alpha_1\alpha_2)$.

We define $T(s, \delta_1, \delta_2)$ as the number of Eichler orders of level N containing $\mathcal{O}_{1,2}$.

Then, there exists a constant C such that :

$$T(s, \delta_1, \delta_2) \leq C\tau(\Delta_{1,2}/\Delta)\tau(f(s, \delta_1, \delta_2))f(s, \delta_1, \delta_2) \quad (2)$$

where $\tau(x)$ is the number of divisors of x and $f(s, \delta_1, \delta_2)^2 = \gcd(f(\delta_1)^2, f(\delta_2)^2, (1-2s))$ when $\mathrm{tr}(\alpha_1)\mathrm{tr}(\alpha_2) = 1$ and $\gcd(f(\delta_1)^2, f(\delta_2)^2, s)$ otherwise.

Proof. Since α_1, α_2 are supposed to reach successive minimas, it is easy to see that their trace must be either 0 or 1 (if not then there would be an element of smaller than norm inside $\mathbb{Z} + \alpha_i$).

By [18, 24.1.4], there exists a unique integer $f(O_{1,2})$ and Gorenstein order $\mathrm{Gor}(\mathcal{O}_{1,2})$ such that $\mathcal{O}_{1,2} = \mathbb{Z} + f(\mathcal{O}_{1,2})\mathrm{Gor}(\mathcal{O}_{1,2})$, where $f(\mathcal{O}_{1,2})$ is an integer and $\mathrm{Gor}(\mathcal{O}_{1,2})$.

We start by showing that $f(\mathcal{O}_{1,2})^2 = \gcd(f(\delta_1)^2, f(\delta_2)^2, (1-2s))$ when $\mathrm{tr}(\alpha_1)\mathrm{tr}(\alpha_2) = 1$ and $f(\mathcal{O}_{1,2})^2 = \gcd(f(\delta_1)^2, f(\delta_2)^2, s)$ otherwise.

The number $f(\mathcal{O}_{1,2})$ divides all the coefficients of the ternary quadratic form associated to the trace 0 elements of $\mathcal{O}_{1,2}$ (see [18, 24.2]). In particular, this means that $f(\mathcal{O}_{1,2})$ divides the conductor of all imaginary quadratic orders contained in $\mathcal{O}1, 2$. This proves that $f(\mathcal{O}_{1,2})^2$ divides $\gcd(f(\delta_1)^2, f(\delta_2)^2)$.

When $\mathrm{tr}(\alpha_i) = 1$, the conductor must be odd. When $\mathrm{tr}(\alpha_1)\mathrm{tr}(\alpha_2) = 1$ we know that $f(\mathcal{O}_{1,2})$ is odd. With disc $\mathbb{Z} + f\mathcal{O} = f^3\mathrm{disc}\ O$ for any $f, \mathcal{O}$, and Proposition 2, we get that $f(\mathcal{O}_{1,2})^3 \mid (\mathrm{tr}(\alpha_1)\mathrm{tr}(\alpha_2) - 2s)^2$. Thus, $f(\mathcal{O}_{1,2}) \mid (\mathrm{tr}(\alpha_1)\mathrm{tr}(\alpha_2) - 2s)$.

Moreover, since $f(\mathcal{O}_{1,2})$ divides the conductor of $\mathbb{Z}[\alpha_1\alpha_2]$, $f(\mathcal{O}_{1,2})^2$ divides its discriminant which is equal to

$$\begin{aligned} s^2 - 4n(\alpha_1)n(\alpha_2) =& \frac{-\delta_1\delta_2 - 1 + \delta_2 + \delta_1 + 4s^2}{4} \\ =& \frac{-\delta_1\delta_2 + \delta_2 + \delta_1 + (2s-1)(2s+1)}{4} \end{aligned}$$

Since $f(\mathcal{O}_{1,2})^2$ divides δ_1, δ_2 it must divide $(2s-1)(2s+1)$ and since $f(\mathcal{O}_{1,2})$ is odd, it cannot divide $(2s+1)$ as it already divides $2s-1$. Thus, $f(\mathcal{O}_{1,2})^2$ divides $(2s-1)$.

When $\mathrm{tr}(\alpha_1) = 1$ and $\mathrm{tr}(\alpha_2) = 0$, we must have that $f(\mathcal{O}_{1,2})$ is odd and that $f(\mathcal{O}_{1,2})^2$ divides the conductor of $\alpha_1 + \alpha_2$. We have $\mathrm{tr}(\alpha_1 + \alpha_2) = 1$ and $n(\alpha_1 + \alpha_2) = n(\alpha_1) + n(\alpha_2) + \mathrm{tr}(\alpha_1\overline{\alpha_2})$. Since $\mathrm{tr}(\alpha_2) = 0$, $\overline{\alpha_2} = -\alpha_2$ and so $n(\alpha_1 - \alpha_2) = n(\alpha_1) + n(\alpha_2) - s$. With $n(\alpha_1) = (1-\delta_1)/4$ and $n(\alpha_2) = -\delta_2/4$, we get that the discriminant of $\alpha_1 + \alpha_2$ is $\delta_1 + \delta_2 - 4s$. Thus, we must have that $f(\mathcal{O}_{1,2})^2 \mid s$ which proves the result.

A similar reasonning proves the result when $\mathrm{tr}(\alpha_1) = 0$ and $\mathrm{tr}(\alpha_2) = 0$.

Now we need to prove that $\gcd(f(\delta_1)^2, f(\delta_2)^2, (\mathrm{tr}(\alpha_1)\mathrm{tr}(\alpha_2) - 2s))$ must divide $f(\mathcal{O}_{1,2})^2$, and the same with $\gcd(f(\delta_1)^2, f(\delta_2)^2, s)$.

$f(\mathcal{O}_{1,2})^2$ is the gcd of the norms of all the trace 0 element in $\mathcal{O}_{1,2}$. By expressing the ternary quadratic form corresponding to the norm of trace 0 elements given as alinear combinations of $1, \alpha_1, \alpha_2, \alpha_1\alpha_2$, it is easy to verify that $\gcd(f(\delta_1)^2, f(\delta_2)^2, (1-2s))$ (resp. $\gcd(f(\delta_1)^2, f(\delta_2)^2, s)$) when $\mathrm{tr}(\alpha_1)\mathrm{tr}(\alpha_2) = 1$ (resp. otherwise) divides the norm of all the elements of trace 0 in $\mathcal{O}_{1,2}$. This proves the result.

Then, we show that $\mathrm{Gor}(\mathcal{O}_{1,2})$ is a Bass order. A Gorenstein order is Bass if all its superorder are Gorenstein. Thus, if $\mathrm{Gor}(\mathcal{O}_{1,2})$ is not a Bass order, there exists a non-Gorenstein order $\mathcal{O}' = \mathbb{Z} + f(\mathcal{O}')\mathrm{Gor}(\mathcal{O}')$ that contains $\mathrm{Gor}\mathcal{O}_{1,2}$. Since $\mathcal{O}'$ is non-Gorenstein, we must have $f(\mathcal{O}') > 1$. We can show that $\mathrm{Gor}\mathcal{O}_{1,2}$ contains the quadratic imaginary orders of $\mathcal{O}_{1,2}$, but with a conductor divided by $f(\mathcal{O}_{1,2})$. We can follow the same reasoning we just led to prove that $f(\mathcal{O}')^2$ must divide $\gcd(f(\delta_1)^2, f(\delta_2)^2, (\mathrm{tr}(\alpha_1)\mathrm{tr}(\alpha_2) - 2s))/(f(\mathcal{O}_{1,2})^2)$ and this value is 1 which is a contradiction.

Thus $\mathrm{Gor}\mathcal{O}_{1,2}$ is Bass.

Eichler and Brzezinski proved that the number of Eichler orders of discriminant Δ containing a given Bass suborder of discriminant $\Delta \mid D$ (in fact their result is about the number of maximal orders containing some Bass sub-order, but it can easily be extended to Eichler orders of level N) is upper-bound by $\tau D/\Delta$ where τ is the function counting the number of divisors of any given number.

To conclude our proof, we just need a result to quantify the number of quaternion order of discriminant disc $\mathcal{O}$ containing a given order of the form $\mathbb{Z} + f\mathcal{O}$ for any integer f and Bass order $\mathcal{O}$. Leroux [13, Lemma 3] provided such a result when f is prime and $\mathcal{O}$ is a maximal order. We will adapt his proof to show that if $\mathbb{Z} + f\mathcal{O}$ is contained in $\mathcal{O}'$ where $\mathcal{O}$ and $\mathcal{O}'$ are Bass orders of the same discriminant, then $\mathcal{O}$ and $\mathcal{O}'$ are connected with an primitive ideal of norm $f' \mid f$.

Let us consider the ideal $I = \{x \in \mathcal{O}', x\mathcal{O} \subset \mathcal{O}'\}$. It is easily verified that this is an integral ideal whose left order is $\mathcal{O}'$ and right order is $\mathcal{O}$. We have $f\mathcal{O}' \subset I \subset \mathcal{O}'$, and so I is equal to $f_1 I'$ where f_1 and $n(I')$ divide f and I' is a primitive integral ideal whose left order is $\mathcal{O}'$ and right order is $\mathcal{O}$.

Thus, we can bound the number $\mathcal{O}'$ of orders containing $\mathbb{Z}+f\mathcal{O}$ with disc $\mathcal{O}' =$ disc $\mathcal{O}$ by $C\tau(f)f$ for some constant C as the number of integral ideals of norm f is in $O(f)$.

We get the final results by multiplying the bound on the number of super-order containing the Bass order $\mathrm{Gor}\mathcal{O}_{1,2}$ with $Cf(\mathcal{O}_{1,2})\tau(f_{\mathcal{O}_{1,2}})$.

□

Proposition 4. *Let $p, t, n, \mathcal{O}$ be such that there exists an element of trace t and norm n inside $\mathcal{O}$ and $n > p^{4/3}$,* MaximalOrderEmbeddingEichler *will output an element α of the correct trace and norm inside $\mathcal{O}$.*

For any $\varepsilon > 0$, the average complexity of MaximalOrderEmbeddingEichler *is*

$$O\left(\frac{n^{3/4+\varepsilon}}{p}\right).$$

Proof. Correctness follow from Proposition 3 and the fact that if N is split in $\mathfrak{O}$, then α of trace t and norm n, is contained in one of the two Eichler orders of the form $\mathbb{Z} + \mathcal{O}\mathfrak{N}$ where $\mathfrak{N}$ is an $\mathfrak{O}$-ideal of norm N.

We are going to prove that the sequential executions of MaximalOrderEmbeddingEichler on all types of maximal orders in $B_{p,\infty}$ takes time $O(n^{3/4+\varepsilon})$. This will prove the result as there are $O(p)$ distinct maximal order types in $B_{p,\infty}$.

Since the values t, n are always the same, the value of N can be the same accross all executions of MaximalOrderEmbeddingEichler. In that case, the sequential executions of

MaximalOrderEmbeddingEichler on all maximal orders types simply consist in the computation of all Eichler orders of level N, and the sequential executions of GenericOrderEmbedding on all these orders.

The integer N is prime, so there are $O(Np)$ Eichlers orders of level N, and each one can be computed in $O(\text{polylog}(pN))$ by enumerating ideals of norm N and intersecting their left and right orders. With the choice of N, the cost of computing them all is $O(n^{3/4}\text{polylog}(pn))$.

Let us write $\mathfrak{S}_{N,p}$ the set of all Eichler orders of level N in $B_{p,\infty}$ and let us write $pN = \varDelta$ the discriminant of these orders. For each $\mathcal{O} \in \mathfrak{S}_{N,p}$, the cost of executing GenericOrderEmbedding on input $\mathcal{O}, t, n$ is written $C_{\mathcal{O}}$. We write $n_1^{\mathcal{O}}, n_2^{\mathcal{O}}, n_3^{\mathcal{O}}$ the norm of the successive minima. Corollary 4 proves that there exists a function $C : \mathbb{N}^4 \to \mathbb{N}$ such that $C_{\mathcal{O}} = O(C(\varDelta, n, n_1^{\mathcal{O}}, n_2^{\mathcal{O}}))$.

By Lemma 1, we have that $n_1^{\mathcal{O}} \leq 2\varDelta^{2/3}$, and by Lemmas 2 and 3 we have that $\max(\varDelta/(4n_1^{\mathcal{O}}), n_1^{\mathcal{O}}) \leq n_2^{\mathcal{O}} \leq 2\sqrt{2}\varDelta/\sqrt{n_1^{\mathcal{O}}}$.

Now, let us define $\delta_i^{\mathcal{O}}$ as the discriminant of $\mathbb{Z}[\beta_i^{\mathcal{O}}]$. Its value is $((\varepsilon_i^{\mathcal{O}})^2 - 4n_i^{\mathcal{O}})$ where $\varepsilon_i^{\mathcal{O}} = \text{tr}(\beta_i^{\mathcal{O}})$ is a value in $\{0,1\}$. In particular, we have $4n_i^{\mathcal{O}} - 1 \leq -\delta_i^{\mathcal{O}} \leq 4n_i^{\mathcal{O}}$. In that case, note that we also have $C_{\mathcal{O}} = O(C(\varDelta, n, -\delta_1^{\mathcal{O}}, -\delta_2^{\mathcal{O}}))$

If we write $T(\delta_1, \delta_2) = \#\ \{\mathcal{O} \in \mathfrak{S}_{N,p} |\ \delta_1^{\mathcal{O}} = \delta_1, \delta_2^{\mathcal{O}} = n_2\}$, and we regroup maximal orders by the discriminants corresponding to their first and second successive minimas, we can get

$$\sum_{\mathcal{O}\in\mathfrak{S}_{N,p}} C_{\mathfrak{O}} \leq C_1 \sum_{-\delta_1=3}^{\lceil 8\varDelta^{2/3}\rceil} \sum_{-\delta_2=\lfloor \max(4\varDelta/(-\delta_1),-\delta_1)\rfloor}^{\lceil \sqrt{2}\varDelta/\sqrt{-\delta_1}\rceil} T(\delta_1,\delta_2)C(\varDelta,n,-\delta_1,-\delta_2) \tag{3}$$

for some constant C_1.

When $\delta_2 \geq \varDelta^2/(16n) - 1$, the bound (ii) from Corollary 4 yields

$$\begin{aligned}
&\sum_{-\delta_1=3}^{\lceil 8\varDelta^{2/3}\rceil} \sum_{-\delta_2=\lceil \varDelta^2/(4n)\rceil}^{\lceil \sqrt{2}\varDelta/\sqrt{-\delta_1}\rceil} T(\delta_1,\delta_2)C(\varDelta,n,-\delta_1,-\delta_2)\\
\leq& C_2 \sum_{-\delta_1=3}^{\lceil 8\varDelta^{2/3}\rceil} \frac{n}{\varDelta\sqrt{-\delta_1}} \sum_{-\delta_2=\lceil \varDelta^2/(n)\rceil}^{\lceil \sqrt{2}\varDelta/\sqrt{-\delta_1}\rceil} T(\delta_1,\delta_2)
\end{aligned}$$

for some constant C_2.

We have an optimal embedding of the quadratic order of discriminant δ_1 inside every Eichler order such that $\delta_1^{\mathcal{O}} = \delta_1$. Let us write $\mathfrak{O}_1$ for this quadratic order.

Each optimal embedding of $\mathfrak{O}_1$ inside an Eichler order $\mathcal{O}$ of level N gives an optimal embedding of $\mathfrak{O}_1$ in the two maximal super-orders of $\mathcal{O}$. There are $O(h(\mathfrak{O}_1))$ distinct optimal embeddings of $\mathfrak{O}_1$ inside maximal orders (see [15, Proposition 3.3] for instance) and it can be shown that each of these embeddings gives an embedding of $\mathfrak{O}_1$ in at most 2 Eichler orders of level N (corresponding to the at most 2 $\mathfrak{O}_1$-ideals of norm N).

Thus, there are $O(h(\mathfrak{O}_1))$ distinct types of Eichler orders of level N with $\delta_1^{\mathcal{O}} = \delta_1$ and we deduce that $\sum_{-\delta_2=\lceil \Delta^2/(4n)\rceil}^{\lceil \sqrt{2}\Delta/\sqrt{-\delta_1}\rceil} T(\delta_1,\delta_2) = O(h(\mathfrak{O}_1)) = O((-\delta_1)^{1/2+\varepsilon})$.

With $\Delta = pN = O(n^{3/4})$, we deduce

$$\sum_{-\delta_1=3}^{\lceil 8\Delta^{2/3}\rceil} \sum_{-\delta_2=\lceil \Delta^2/(16n)\rceil}^{\lceil \sqrt{2}\Delta/\sqrt{-\delta_1}\rceil} T(\delta_1,\delta_2)C(\Delta,n,-\delta_1,-\delta_2) = O(n\Delta^{-1/3+\varepsilon}) = O(n^{3/4+\varepsilon}) \tag{4}$$

For every Eichler order $\mathcal{O}$ containing an embedding of the quadratic orders of discriminant δ_1, δ_2, Proposition 2 tells us that their must be a value $s = (1/2) \pm \sqrt{\delta_1\delta_2} + \varepsilon$ mod Δ where $|s| \leq \sqrt{\delta_1\delta}$. In that case, the value of $\delta_1\delta_2 = (2s-\varepsilon)^2 + k\Delta$ for some integer $0 \leq k$.

Thus we can upper-bound the second part of our sum as follows:

$$\begin{aligned}&\sum_{-\delta_1=3}^{\lceil 8\Delta^{2/3}\rceil} \sum_{-\delta_2=\lfloor \max(4\Delta/(-\delta_1),-\delta_1)\rfloor}^{\lceil \Delta^2/(16n)\rceil} T(\delta_1,\delta_2)C(\Delta,n,-\delta_1,-\delta_2)\\ &\leq \sum_{|s|\leq 2\Delta/\sqrt{n}} \sum_{k=0}^{\lceil \Delta^{5/3}/n\rceil} \sum_{\delta_1,\delta_2\in\{\delta_1,\delta_2|\delta_1\delta_2,\ (2s-\varepsilon)^2+k\Delta\}} T(s,\delta_1,\delta_2)C(\Delta,n,-\delta_1,-\delta_2)\end{aligned}$$

where $T(s,\delta_1,\delta_2)$ was defined in Lemma 7.

Now, we can apply Corollary 4 (i) to get

$$\begin{aligned}&\sum_{|s|\leq 2\Delta/\sqrt{n}} \sum_{k=0}^{\lceil \Delta^{5/3}/n\rceil} \sum_{\delta_1,\delta_2\in\{\delta_1,\delta_2|\delta_1\delta_2,\ (2s-\varepsilon)^2+k\Delta\}} T(s,\delta_1,\delta_2)C(\Delta,n,-\delta_1,-\delta_2)\\ &\leq \sqrt{n} \sum_{|s|\leq 2\Delta/\sqrt{n}} \sum_{k=0}^{\lceil \Delta^{5/3}/n\rceil} \sum_{\delta_1,\delta_2\in\{\delta_1,\delta_2|\delta_1\delta_2,\ (2s-\varepsilon)^2+k\Delta\}} \frac{T(s,\delta_1,\delta_2)}{\sqrt{\delta_1\delta_2}}\end{aligned}$$

We have $\delta_1\delta_2 > s^2$, thus $1/\sqrt{\delta_1\delta_2} \leq 1/|s|$. Moreover, we can apply Lemma 7 to upper-bound $T(s,\delta_1,\delta_2)$. This yields

$$\begin{aligned}
&\sum_{|s|\leq 2\Delta/\sqrt{n}} \sum_{k=0}^{\lceil \Delta^{5/3}/n\rceil} \sum_{\delta_1,\delta_2\in\{\delta_1,\delta_2|\delta_1\delta_2,\ (2s-\varepsilon)^2+k\Delta\}} \frac{T(s,\delta_1,\delta_2)}{\sqrt{\delta_1\delta_2}}\\
\leq&\sqrt{n}\max_{x\leq n^m}\tau(N)^2 \sum_{|s|\leq 2\Delta/\sqrt{n}} \frac{1}{|s|} \sum_{k=0}^{\lceil \Delta^{5/3}/n\rceil} \sum_{\delta_1,\delta_2\in\{\delta_1,\delta_2|\delta_1\delta_2,\ (2s-\varepsilon)^2+k\Delta\}} f(s,\delta_1,\delta_2)
\end{aligned}$$

for some constant C_3 and integer $m > 0$. We define $\tau(x)$ to be the number of distinct divisor of any integer x.

The size of the set $\{\delta_1,\delta_2 \mid \delta_1\delta_2,\ (2s-\varepsilon)^2+k\Delta\}$ can be uppper-bounded by $\tau((2s-\varepsilon)^2+k\Delta)^2$.

By definition of $f(s,\delta_1,\delta_2)$ in Lemma 7, we see that we must have $f(s,\delta_1,\delta_2)^4 \mid k$. Thus, by writing every value k as $k_0^4k_1$ we can upper bound $f(s,\delta_1,\delta_2)^4$ by k_0, and we obtain :

$$\begin{aligned}
&\sqrt{n}\max_{x\leq n^m}\tau(N)^2 \sum_{|s|\leq 2\Delta/\sqrt{n}} \frac{1}{|s|} \sum_{k=0}^{\lceil \Delta^{5/3}/n\rceil} \sum_{\delta_1,\delta_2\in\{\delta_1,\delta_2|\delta_1\delta_2,\ (2s-\varepsilon)^2+k\Delta\}} f(s,\delta_1,\delta_2)\\
\leq&\sqrt{n}\max_{x\leq n^m}\tau(N)^4 \sum_{|s|\leq 2\Delta/\sqrt{n}} \frac{1}{|s|} \sum_{k_0=1}^{\lceil (\Delta^{5/3}/n)^{1/4}\rceil} \sum_{k_1=0}^{\lfloor \Delta^{5/3}/(nk_0^4)\rfloor} k_0
\end{aligned}$$

There exists a constant C_4 such that

$$\sum_{k_0=1}^{\lceil (\Delta^{5/3}/n)^{1/4}\rceil} \sum_{k_1=0}^{\lfloor \Delta^{5/3}/(nk_0^4)\rfloor} k_0 \leq C_4\Delta^{5/3}/n \sum_{k_0=1}^{\lceil (\Delta^{5/3}/n)^{1/4}\rceil} \frac{1}{k_0^3}.$$

The value of $\zeta(3)$ is a constant and $\sum_{s=1}^{x} 1/s = O(\log x)$. Thus, there is a constant C_5 such that

$$\begin{aligned}
&\sqrt{n}\max_{x\leq n^m}\tau(N)^4 \sum_{|s|\leq 2\Delta/\sqrt{n}} \frac{1}{|s|} \sum_{k_0=1}^{\lceil (\Delta^{5/3}/n)^{1/4}\rceil} \sum_{k_1=0}^{\lfloor \Delta^{5/3}/(nk_0^4)\rfloor} k_0\\
\leq&C_5\frac{\Delta^{5/3}}{\sqrt{n}}\log(\Delta/\sqrt{n})\max_{x\leq n^m}\tau(N)^4
\end{aligned}$$

With $\Delta = pN = O(n^{3/4})$ and the fact that $\tau(x) = O(x^{\varepsilon})$ for any $\varepsilon > 0$ [20], we conclude that

$$\sum_{-\delta_1=3}^{\lceil 8\Delta^{2/3}\rceil} \sum_{-\delta_2=\lfloor \max(4\Delta/(-\delta_1),-\delta_1)\rfloor}^{\lceil \Delta^2/(16n)\rceil} T(\delta_1,\delta_2)C(\Delta,n,-\delta_1,-\delta_2) = O(n^{3/4+\varepsilon}) \tag{5}$$

The combination of Eqs. (3) to (5) proves that executing MaximalOrderEmbeddingEichler on all maximal orders takes time $O(n^{3/4+\varepsilon})$ and this proves that the average running time is $O(n^{3/4+\varepsilon})/p$. □

3.3 Another Heuristic Algorithm with Factorization

The problem with GenericOrderEmbedding is that it does not work well when the input order $\mathcal{O}$ contains smaller elements that one should expect from a random order of the same discriminant. Thus, while being efficient in the average case, it is not always optimal. Interestingly, we will see that the bad cases for GenericOrderEmbedding are actually good cases for another algorithm that we present below as GenericOrderEmbeddingFactorization.

The idea of this algorithm is that since $\mathcal{O}$ contains a very small element β_1, it will be easier to know the value of $\mathrm{tr}(\alpha\beta_1)$ exactly. Then, once this value is fixed, the ternary quadratic form becomes a binary quadratic form that we know how to solve efficiently.

More precicely, let β be an element in $\mathcal{O}$ for which we know $\mathrm{tr}(\alpha\beta)$, and let γ be any element in $\mathcal{O}$ orthogonal to $\mathbb{Z}[\beta]$. Now look at the order

$$\mathbb{Z}\langle\beta,\gamma\rangle \subseteq \mathcal{O}$$

and write M for the index $[\mathbb{Z}\langle\beta,\gamma\rangle : \mathcal{O}]$. Writing $x+\beta y+\gamma z+\gamma\beta w$ for a generic element in $\mathbb{Z}\langle\beta,\gamma\rangle$, the norm form of this order is of the simple form

$$Q(x,y,z,w) := f(x,y) + \mathrm{n}(\gamma) f(z,w)$$

where $f(x,y)$ denotes the norm of $x+\beta y$. While it is unlikely that α lies in $\mathbb{Z}\langle\beta,\gamma\rangle$, we have that $M\alpha \in \mathbb{Z}\langle\beta,\gamma\rangle$, thus, we can instead solve for $M\alpha$. First, we find the values of x and y from the knowledge of $\mathrm{tr}(\alpha)$ and $\mathrm{tr}(\alpha\beta)$ (because z, w contribute nothing to these traces). Then, we can solve for z, w by enumerating all solutions of

$$f(z,w) = \frac{\mathrm{n}(M\alpha) - \mathrm{n}(\alpha_0)}{\mathrm{n}(\gamma)}.$$

with Cornacchia's algorithm. Finally, for each potential solution of the form $\alpha' := x+\beta y+\gamma z+\gamma\beta w$, we check if $\alpha'/M \in \mathcal{O}$.

The only caveat is that Cornacchia's algorithm require the factorization of the number one is trying to represent. Furthermore, the total amount of solutions is exponential in the number of distinct prime factors of the number one is trying to represent. Thus, the best we can do, as we need to enumerate through all the solutions of each Cornacchia instance, is get a heuristic runtime for our algorithm, under the plausible assumption that the integers that we encounter will not have too many prime factors.

Heuristic 1 *All integers M (resp. N) occuring in Step 4 (resp. Step 8) of Algorithm 3 behave like four times random numbers (resp. random numbers). In particular, the number of distinct prime factors is exptected to be small, i.e. $O(poly(\log\log M))$ (resp. $O(poly(\log\log N))$) .*

We also introduce a second heuristic to estimate the number of expected embedding of a given quadratic order in any maximal order. This heuristic will be useful in both the proof of this algorithm and later.

Heuristic 2 *Let $\mathcal{O} \subseteq B_{p,\infty}$ be a maximal quaternion order, and let $\mathfrak{O}$ be a quadratic order, embedding into $B_{p,\infty}$. The expected number of optimal embeddings $\iota : \mathfrak{O} \hookrightarrow \mathcal{O}$ up to conjugation by $\mathcal{O}^\times$ is $\Theta(h(\mathfrak{O})/p)$.*

One reasoning for this heuristic comes from [18, Theorem 30.7.5], which, specialized to our case, say that summing over a representative of all isomorphism classes of maximal orders in $B_{p,\infty}$, there should be $\Theta(h(\mathfrak{O}))$ embeddings. Heuristic 2 simply says that these embeddings are randomly distributed over these representatives. In [12], Leroux proved some bounds on the number of distinct embeddings of the same quadratic order inside the same maximal order. While these bounds are not enough to prove Heuristic 2, they are a first step in the right direction as they prove that extreme situations where all quadratic orders are embedded inside the same maximal order are not possible.

Proposition 5. *Assume the existence of a factorization oracle, and that Heuristic 1 holds. Given integers t and n,* GenericOrderEmbeddingFactorization *outputs an element $\alpha \in \mathcal{O}$ with $\operatorname{tr}(\alpha) = t$ and $\operatorname{n}(\alpha) = n$ or decide that none exists in time*

$$O\left(\frac{\sqrt{n\operatorname{n}(\beta_1)}}{\Delta} \cdot \operatorname{polylog}(np)\right)$$

where $\Delta = \operatorname{disc} \mathcal{O}$. Further, assuming $\mathcal{O}$ is maximal, and that Heuristic 2 holds, the expected runtime is also upper bounded by

$$O\left(\sqrt{\operatorname{n}(\beta_1)} \cdot \operatorname{polylog}(np)\right) \subseteq O\left(p^{1/3} \cdot \operatorname{polylog}(np)\right)$$

Proof. The correctness of the algorithm follows directly from the description at the start of this section. We now proceed to prove the runtime of the algorithm.

The algorithm tries the $O\left(\frac{\sqrt{n\operatorname{n}(\beta_1)}}{\Delta}\right)$ possible values of t_1, and for each one, attempts to derive a solution α from representations of some integer MN by the principal binary quadratic form corresponding to elements in $\mathbb{Z}[\beta_1]$.

Under Heuristic 1, Cornacchia's algorithm can find the $O(\operatorname{polylog}(MN))$ solutions in $O(\operatorname{polylog}(MN))$ time. Testing each candidate has the same complexity and this proves the first part of the result.

Algorithm 3 GenericOrderEmbeddingFactorization$(\mathcal{O}, t, n)$

```
Input: An order $\mathcal{O} \subset B_{p,\infty}$ of discrd $\mathcal{O} = \Delta$ (with known factorization), and two integers
    $t, n \in \mathbb{Z}$.
Output: $\perp$, or $\alpha \in \mathcal{O}$ with $n(\alpha) = n$ and $\mathrm{tr}(\alpha) = t$.
 1: Compute a Minkowski reduced basis $1, \beta_1, \beta_2, \beta_3$ of $\mathcal{O}$.
 2: Compute $D_1 = \mathrm{tr}(\beta_1)^2 - 4n(\beta_1)$, and $D = t^2 - 4n$.
 3: Compute $\gamma_1, \gamma_2$, a Minkowski reduced basis of the part of $\mathcal{O}$ orthogonal to $\mathbb{Z}[\beta]$.
 4: Compute $M := [\mathbb{Z}\langle \beta_1, \gamma \rangle : \mathcal{O}]$.
 5: Compute $s_1$ a square root of $DD_1 \mod \Delta$.
 6: for $t_1 \in [1, \sqrt{4nn(\beta_1)}]$ such that $t_1 = (1/2)(\pm s_1 + t\mathrm{tr}(\beta_1)) \mod \Delta$ do
 7:     Let $\alpha_0' := x + y\beta_1$ be the element in $\mathbb{Z}[\beta_1]$ with $\mathrm{tr}(\alpha_0') = Mt, \mathrm{tr}(\alpha_0'\beta_1) = Mt_1$
 8:     Set $N := \frac{M^2 n - \mathrm{n}(\alpha_0)}{M\mathrm{n}(\gamma_1)}$
 9:     for $z, w$ such that $\mathrm{n}(z + \beta_1 w) = MN$ do
10:         Set $\alpha' := \alpha_0' + \gamma(z + \beta_1 w)$
11:         if $\alpha'/M \in \mathcal{O}$ then
12:             return $\alpha'/M$
13:         end if
14:     end for
15: end for
16: return Return $\perp$.
```

However, the runtime above is the same as the time it takes to find all solutions. Applying Heuristic 2, we expect there to be a total of

$$O(h(\mathbb{Z}[\alpha])/p) = O(\sqrt{n}/p)$$

solutions. By Heuristic 1, each value of t_1 is only expected to give a polylogarithmic number of solutions, hence the total number of values t_1 that corresponds to a value of $\mathrm{tr}(\alpha\beta_1)$ for a solution α, divided by the total possible number of values of t_1, is

$$\tilde{O}\left(\frac{\sqrt{n}/p}{\sqrt{n\mathrm{n}(\beta_1)}/p}\right) = \tilde{O}\left(\frac{1}{\sqrt{\mathrm{n}(\beta_1)}}\right)$$

This bounds the expected number of values of t_1 we have to try before a solution will be found. The final expected runtime is obtained by the bound on $\mathrm{n}(\beta_1)$ given by Lemma 1 □

From Proposition 5, we see that as n increases, the algorithm's runtime eventually becomes independent of n. In the cases when n is very large, we can also discard the factorization oracle altogether, by only running Cornacchia on "easy" instances (for instance when N is a prime number). Indeed, under Heuristic 1, the numbers N in Step 8 behave like random integers of the same size and so they have a probability of $1/\log N$ to be prime. This allow us to run GenericOrderEmbeddingFactorization without the need of a factorization oracle, and the running time is only increased by a factor $O(\log(np))$.

Remark 1. Our algorithm GenericOrderEmbeddingFactorization can be seen as a generalization of the method introduced in [1]. Indeed, what is done in [1, Algorithm 5.1] is equivalent to looking at $\mathrm{tr}(N_0\omega_0\alpha)$ where ω_0 is an integral element of very small norm, and the integer N_0 is such that $\mathcal{O}$ is connected with $\mathcal{O}_0$, a maximal order containing ω_0 by an ideal of norm N_0. Since one can expect $N_0 \approx \sqrt{p}$ when $\mathcal{O}$ is a random maximal order, this method allows us to recover α in polynomial time when $n = O(p)$. In the case where N_0 is especially small, $D\omega_0$ might be equal to β_1 and in that case, our method is equivalent to the one of [1]. Note that in every other case, our method is strictly better. Also, note that, by replacing β_1 by other elements of small norm, we can perform a similar randomization as was explained in [1, Section 5.3], to remove the need for the factorization oracle. However, this rerandomization may not help in some cases where β_1 is much smaller than β_2, β_3, because in that case all the small vectors will lie in the same quadratic order generated by 1 and β_1.

4 Ideals Between Oriented Orders

In this section, we expand on results related to primitively oriented maximal orders. We do this by first considering ideals between oriented maximal orders, and show that such ideals "comes from" quadratic ideals precicely when the left and right order of the ideal admits the same orientation.

Our first result is then a new algorithm that reduces $\mathfrak{O}$-vectorisation to $\mathfrak{O}$-EndRing in polynomial time for all orders. One such reduction first appeared in the work by Castryck, Vercauteren and Panny [4] for the order $\mathbb{Z}[\sqrt{-p}]$, and this was later generalised Wesolowski [19] to arbitrary orders. However, the algorithm is only polynomial in $\#\mathrm{Cl}(\mathfrak{O})[2]$ (which can be exponential in the discriminant of $\mathfrak{O}$), and requires the factorization of disc $\mathfrak{O}$. Our reduction does not have this caveat, and only depends on the size of the discriminant, not its factorization.

Second, we consider the problem of finding ideals of fixed degree between isomorphism classes of quaternion orders, a problem of huge importance in isogeny-based cryptography. For large norm, this is solved efficiently by the (generalised) KLPT algorithm [11] [6], while for small degree, this is efficiently solvable by simple lattice reduction. The remaining sizes of norms in between here were recently studied by Benjamin Bencina, Péter Kutas, Simon-Philipp Merz, Christophe Petit, Miha Stopar and Charlotte Weitkämper [2]. The relation to the quaternion embedding problem was mentioned in the same work [2, Appendix A]. We expand on this connection, giving a heuristic algorithm which solves this problem in time $O(p^{2/3})$ for any degree d, and we show that in the special case where both orders are oriented orders of small class number, this algorithm is polynomial time for any d, allowing the computation of optimal paths between supersingular curves with small endomorphisms. Finally, in Appendix C, we consider the case where one of the orders contain an element of very small norm, and show that this can be solved in polynomial time up to $d < p^{2/3}$.

Given an $\mathfrak{O}$-ideal $\mathfrak{l}$, and an primitively $\mathfrak{O}$-oriented order $(\mathcal{O}, \iota)$ (Definition 1) we can define the corresponding quaternion ideal as $\mathcal{O}\langle\iota(\mathfrak{l})\rangle$. Further, given an $\mathcal{O}$-ideal I, one can define the corresponding $\mathfrak{O}$-ideal to be $\iota^{-1}(I)$, which can be computed by intersecting I with $\iota(K)$. The relation between these operations are given by the following proposition.

Proposition 6. *Let $(\mathcal{O}, \iota)$ be a primitively $\mathfrak{O}$-oriented order. Then*

- *Given a left $\mathcal{O}$-ideal I, we have that $\mathcal{O}\langle \mathrm{n}(I)\rangle \subseteq \mathcal{O}\langle I \cap \iota(K)\rangle \subseteq I$.*
- *Given an invertible $\mathfrak{O}$-ideal $\mathfrak{l}$, we have that $\mathcal{O}\langle\iota(\mathfrak{l})\rangle \cap \iota(K) = \iota(\mathfrak{l})$*

Proof. To prove the first statement, note that the first inequality follows from the fact that $\mathrm{n}(I)\mathbb{Z} \subseteq I \cap \iota(\mathfrak{O})$, and the second follows from the observation that $O\langle I \cap \iota(\mathfrak{O})\rangle \subseteq I$.

To prove the second statement, following [18, Exercise 30.2.a], we see that $\mathcal{O}\langle\iota(\mathfrak{l})\rangle \cap \iota(K) \supset \iota(\mathfrak{l})$, since $1 \in \mathcal{O}$, and conversely, since $\mathfrak{l}$ is invertible, we have find that

$$\begin{aligned}(\mathcal{O}\langle\iota(\mathfrak{l})\rangle \cap \iota(K))\iota(\mathfrak{O}) &= (\mathcal{O}\langle\iota(\mathfrak{l})\rangle \cap \iota(K))\iota(\mathfrak{l}^{-1}\mathfrak{l}) \\ &\subseteq (\mathcal{O}\langle\iota(\mathfrak{l})\iota(\mathfrak{l}^{-1})\rangle \cap \iota(K))\iota(\mathfrak{l}) = \iota(\mathfrak{O})\mathfrak{l}\end{aligned}$$

where we are using $\mathcal{O}\langle\iota(\mathfrak{l})\iota(\mathfrak{l}^{-1})\rangle \cap \iota(K) = \mathcal{O} \cap \iota(K) = \iota(\mathfrak{O})$, which follows by definition of $(\mathcal{O}, \iota)$ being primitively $\mathfrak{O}$-oriented. □

The previous proposition motivates the following definition, which emphasises when a quaternion ideal is generated by the image of a quadratic ideal:

Definition 2. *Let $(\mathcal{O}, \iota)$ be a primitively $\mathfrak{O}$-oriented maximal order. A left $\mathcal{O}$-ideal is said to be generated by an $\mathfrak{O}$-ideal if*

$$I = \mathcal{O}\langle I \cap \iota(K)\rangle$$

The following lemma shows that the orientation automatically "transfer" to the right order of an ideal generated by an $\mathfrak{O}$-ideal.

Lemma 8. *Let $(\mathcal{O}, \iota)$ be a primitively $\mathfrak{O}$-oriented maximal order, and let I be a left $\mathcal{O}$-ideal, generated by an $\mathfrak{O}$-ideal. Then $(\mathcal{O}_R(I), \iota)$ is a (not necessarily primitively) $\mathfrak{O}$-oriented maximal order.*

Proof. Let ω be the image of a generator of $\mathfrak{O}$ under ι. To prove that $(\mathcal{O}_R(I), \iota)$ is a $\mathfrak{O}$-oriented maximal order, it suffices to see that $\omega \in \mathcal{O}_R(I)$. But this follows from the fact that I can be given generators in $\iota(K)$, which commute with ω. □

When we have a primitively $\mathfrak{O}$-oriented maximal order $(\mathcal{O}, \iota)$, the previous lemma showed that given an $\mathcal{O}$-ideal I, the right order of I admitting the same orientation is a necessary condition for I to be generated by an $\mathfrak{O}$-ideal. Next, we show that this condition is also sufficient.

Lemma 9. *Let $(\mathcal{O}_1, \iota)$ and primitively $\mathfrak{O}$-oriented maximal order, and let $(\mathcal{O}_2, \iota)$ be a (not necessarily primitively) $\mathfrak{O}$-oriented maximal order. Then their connecting ideal I is generated by an $\mathfrak{O}$-ideal.*

Proof. Let ω be a generator of $\mathfrak{O}$ under ι, and let I be the unique primitive connecting ideal between $\mathcal{O}_1, \mathcal{O}_2$. We have that $\omega \in \mathcal{O}_1 \cap \mathcal{O}_2 = \mathbb{Z} + I$, and hence, $a + \omega \in I$ for some $a \in \mathbb{Z}$. Let

$$J = \mathcal{O}_1 \langle a + \omega, \mathrm{n}(I) \rangle.$$

Clearly, J is generated by an $\mathfrak{O}$-ideal, and we will show that $I = J$. First, note that $\mathcal{O}\langle \mathrm{n}(I) \rangle \subseteq J \subseteq I$, so assume $\mathrm{n}(J) = \mathrm{n}(I)d$ for some $d \mid \mathrm{n}(I)$. Assume now that $J \subsetneq I$, i.e. that $d \neq 1$. Since $\mathcal{O}_1\langle \mathrm{n}(I) \rangle \subseteq J$, we have $\mathcal{O}_1\langle d \rangle \subset J + \mathcal{O}_1\langle d \rangle$. Since we have $\mathrm{n}(J + \mathcal{O}_1\langle d \rangle) = \gcd(n(J), d^2)$, we see that we must have $\mathrm{n}(J + \mathcal{O}_1\langle d \rangle) = d^2$. By equality of the norm, we must have $J + \mathcal{O}_1\langle d \rangle = \mathcal{O}_1\langle d \rangle$. Hence, $J/(d) \subseteq \mathcal{O}_1$, implying that $\frac{a+\omega}{d} \in \mathcal{O}_1$, contradicting the assumption that $\mathcal{O}_1$ was primitively $\mathfrak{O}$-oriented. Hence $I = J$, which shows that I is generated by an $\mathfrak{O}$-ideal. □

4.1 Vectorisation to Oriented Endring Reduction

From Lemma 9, we see that the only obstruction in finding an ideal generated by an $\mathfrak{O}$-ideal between two primitively oriented maximal orders is that the two $\mathfrak{O}$-oriented orders might be oriented in different ways. Fortunately, the following lemma shows that this is easy to fix.

Lemma 10. *Let $(\mathcal{O}, \iota_1)$ be a primitively $\mathfrak{O}$-oriented maximal order, and let $\iota_2 : K \hookrightarrow B$ be another embedding. Then there exists an order $\mathcal{O}' \cong \mathcal{O}$ such that $(\mathcal{O}', \iota_2)$ is a primitively $\mathfrak{O}$-oriented maximal order*

Proof. By the Skolem-Noether theorem, given any two embeddings $\iota_1, \iota_2 : K \hookrightarrow B$, there exists some $\alpha \in B^\times$ such that for any $\delta \in K$, we have that $\iota_1(\delta) = \alpha^{-1}\iota_2(\delta)\alpha$. Then $\mathcal{O}' := \alpha\mathcal{O}\alpha^{-1}$ is isomorphic to $\mathcal{O}$, and further it is clear that

$$\iota_2(K) \cap \mathcal{O}' = \iota_1(K) \cap \alpha\mathcal{O}\alpha^{-1} = \alpha^{-1}\iota_2(K)\alpha \cap \mathcal{O} = \iota_1(K) \cap \mathcal{O} = \iota(\mathfrak{O}),$$

hence $(\mathcal{O}', \iota_2)$ is a primitively oriented maximal order. □

Thus, the reduction simply consists of fixing the orientations, and intersecting with the image of K under the orientation. We summarize this in Algorithm 4.

Algorithm 4 Vectorization$_{\mathfrak{O}}((\mathcal{O}_1, \iota_1), (\mathcal{O}_2, \iota_2))$

```
Input: Two primitively $\mathfrak{O}$-oriented maximal orders $(\mathcal{O}_1, \iota_1), (\mathcal{O}_2, \iota_2)$.
Output: An $\mathfrak{O}$-ideal $\mathfrak{l}$ such that $\mathcal{O}_R(\mathcal{O}_1\langle\mathfrak{l}\rangle) \cong \mathcal{O}_2$.
  Set $\omega_i := \iota_i(\omega)$ for $i = 1, 2$, where $\omega$ is any generator of $\mathfrak{O}$.
  Compute $\alpha$ such that $\alpha\omega_1 - \omega_2\alpha = 0$ using linear algebra.
  Set $\mathcal{O}_2' := \alpha\mathcal{O}_2\alpha^{-1}$.
  Compute the connecting $(\mathcal{O}_1, \mathcal{O}_2')$-ideal $I := N\mathcal{O}_1\mathcal{O}_2'$.
  Set $\mathfrak{l} := \iota^{-1}(I \cap \iota(\mathfrak{O}))$.
  return $\mathfrak{l}$.
```

Proposition 7. *Algorithm 4 is correct and runs in polynomial time in the length of the input.*

Proof. Lemma 10 shows both the existence of α, and that $(\mathcal{O}_2', \iota_1)$ is a primitively $\mathfrak{O}$-oriented order. Thus, it follows from Lemma 9, that the connecting ideal between $\mathcal{O}_1$ and $\mathcal{O}_2'$ is generated by an $\mathfrak{O}$-ideal. Finally, the runtime is clear, as all operations done consists of simple linear algebra. □

The Corollary from Proposition 7 is that $\mathfrak{O}$-Vectorization reduces to $\mathfrak{O}$-Endring in polynomial time, regardless of the size of $\mathrm{Cl}(\mathfrak{O})[2]$, and without knowing the factorization of disc $\mathfrak{O}$, improving the results of Wesolowski [19].

Corollary 5. *Effective $\mathfrak{O}$-Vectorization reduces to $\mathfrak{O}$-Endring in polynomial time.*

Proof. In this proof, we reuse the notation from [19]. We are given two oriented curves $(E_1, \gamma_1), (E_2, \gamma_2) \in SS_{\mathfrak{O}}(p)$, together with an ϵ-basis of $\mathrm{End}(E_1)$ and $\mathrm{End}(E_2)$, and our goal is to compute an $\mathfrak{O}$-ideal $\mathfrak{a}$ such that $\mathfrak{a} \star (E_1, \gamma_1) = (E_2, \gamma_2)$, and an efficient representation of $\phi_{\mathfrak{a}} : (E_1, \gamma_1) \to \mathfrak{a} \star (E_1, \gamma_1)$.

First, compute optimal embeddings ι_1 and ι_2 such that $(\mathrm{End}(E_i), \iota_i)$ are primitively $\mathfrak{O}$-oriented maximal orders using [19, Lemma 2]. Next, we run Algorithm 4 on the primitively oriented orders $(\mathrm{End}(E_1), \iota_1), (\mathrm{End}(E_2), \iota_2)$, which outputs an $\mathfrak{O}$-ideal $\mathfrak{a}$ solving the vectorization problem. Finally, an efficient representation of the isogeny $\phi_{\mathfrak{a}}$ can be computed unconditionally in polynomial time using [16, Theorem 2.8], or, for a more practical alternative, with [19, Proposition 9] assuming GRH. □

4.2 Finding Fixed Norm Ideals Between Maximal Orders

When given two maximal orders $\mathcal{O}_1, \mathcal{O}_2$, we consider the problem of finding a left $\mathcal{O}_1$-ideal I of norm d such that $\mathcal{O}_R(I) \cong \mathcal{O}_2$. This problem is of huge importance in isogeny-based cryptography, as it corresponds to computing isogenies of a given norm between supersingular curves, when they exists. One special case of this, is finding such an ideal of norm ℓ^k for some fixed, small prime ℓ, and the smallest $k \in \mathbb{Z}_{\geq 0}$, such that such an ideal exists. This correspond to an optimal path between the curves in the ℓ-isogeny graph.

In this section, we give a new algorithm for solving this problem, based on our algorithms for the quaternion embedding problem. The algorithm consists of computing the ascending ideal to the correct level, and then bruteforcing the remaining horizontal part, for well chosen embeddings. To do this, we need the following Lemma.

Lemma 11. *Let $(\mathcal{O}, \iota)$ be a primitively $(\mathbb{Z}+d\mathfrak{O})$-oriented maximal order, with $\omega = \iota(d\omega_0)$, where ω_0 is any generator of $\mathfrak{O}$, and let*

$$I := \mathcal{O}\langle \omega, d\rangle.$$

Then, $\mathrm{n}(I) = d$, *and* $(\mathcal{O}_R(I), \iota)$ *is a primitively $\mathfrak{O}$-oriented maximal order.*

Proof. First, note that $\mathrm{n}(I) = d$, because if not, this would contradict the primality of the embedding, by the same argument as the last part of Lemma 9. Next, we show that $(\mathcal{O}_R(I), \iota)$ is a $\mathfrak{O}$-oriented order, i.e. $\omega/d \in \mathcal{O}_R(I)$. To see this, note that for any element

$$\alpha\omega + \beta d \in I, \quad \alpha, \beta \in \mathcal{O}_L(I)$$

we have that

$$\begin{aligned}(\alpha\omega + \beta d)\omega/d &= \alpha\omega^2/d + \beta\omega \\ &= \alpha(\mathrm{tr}(\omega)\omega - \mathrm{n}(\omega))/d + \beta\omega \\ &= \alpha(\mathrm{tr}(\iota(\omega_0))\omega + d\mathrm{n}(\iota(\omega_0))) + \beta\omega \in I\end{aligned}$$

Finally, to see that the $\mathfrak{O}$-embedding on $\mathcal{O}_R(I)$ induced by ι is optimal, note that if it was not, this would again contradict the optimality of the $(\mathbb{Z} + d\mathfrak{O})$-embedding on $\mathcal{O}$, since $\mathrm{n}(I)$ induces the embedding $d\mathcal{O}_R(I) \subseteq \mathcal{O}_L(I)$. □

For simplicity, we will assume factorization, and use a special purpose algorithm we call GenericOrderEmbeddingFactorizationAll, whose only difference with the original GenericOrderEmbeddingFactorization, is that it keeps searching and outputting solutions, until all are found. From the proof of Proposition 5, the expected runtime of this version is still $O\left(\frac{\sqrt{nn(\beta_1)}}{p}\right)$ under Heuristics 1 and 2.

Proposition 8. *Assume the existence of a factorization oracle, and that Heuristic 1 and Heuristic 2 holds. Let β_1 be the smallest non-integer in $\mathcal{O}_1$, and let γ_1 be the smallest non-integer in $\mathcal{O}_2$.* $\mathsf{ConnectingIdealWithNorm}_d$ *always returns a solution I if it exists, or $\perp$ if a solution does not exist, and runs in expected time*

$$O\left(\sqrt{\mathrm{n}(\beta_1)\mathrm{n}(\gamma_1)}\right).$$

Algorithm 5 $\mathsf{ConnectingIdealWithNorm}_d(\mathcal{O}_1, \mathcal{O}_2)$

```
Input: Two maximal orders $\mathcal{O}_1, \mathcal{O}_2 \subset B_{p,\infty}$, and an integer $d$.
Output: $\perp$ or an ideal $I$ with $\mathcal{O}_L(I) = \mathcal{O}_1$, $O_R(I) \cong \mathcal{O}_2$, and $n(I) = d$.
  Let $\gamma_1$ be the element achieveing the first successive minima of $\mathcal{O}_2$.
  for $\omega$ in $\mathsf{GenericOrderEmbeddingFactorizationAll}(\mathcal{O}_1, n(d\gamma_1), t(d\gamma_1))$ do
    Set $d'$ to be the biggest integer s.t. $\omega/d' \in O_1$ for $i = 1, 2, 3$
    Set $I_1 := \mathcal{O}_1\langle \omega/d', d/d' \rangle$
    Set $\mathcal{O}_{\text{crater}} := \mathcal{O}_R(I)$
    Let $\iota : \mathbb{Z}[\gamma_1] \hookrightarrow \mathcal{O}_{\text{crater}}$ be defined by $\iota(\gamma_1) = \omega/d$.
    for $\mathfrak{l}$ in all $\mathbb{Z}[\gamma_1]$-ideals of norm $d'$ do
      Set $I_2 := \mathcal{O}_{\text{crater}}\langle \iota(\mathfrak{l}) \rangle$
      if $\mathcal{O}_R(I_2) \cong \mathcal{O}_2$ then
        return $I_1 \cdot I_2$.
      end if
    end for
  end for
  return $\perp$
```

Proof. First, we prove the correctness of the algorithm. Assume a solution I exists. We will prove that the solution I can be written as product $I = I_1 \cdot I_2$, where I_2 comes from a $\mathbb{Z}[\gamma_1]$-ideal. This will also proves the correctness of the algorithm, as it runs through all embeddings of $\mathbb{Z}[d\gamma_1]$ into $\mathcal{O}_1$, computes the unique corresponding ascending ideal I_1', and then multiplies this with all the remaining ideals that comes from a $\mathbb{Z}[\gamma_1]$-ideals.

Let us denote by ω_0, the element in $\mathcal{O}_R(I)$ such that $\alpha\omega_0\alpha^{-1} = \gamma_1$ for some $\alpha \in B_{p,\infty}^\times$ (this exists since $\mathcal{O}_2 \cong \mathcal{O}_R(I)$). Let $\omega := d\omega_0$. The sequence of inclusions $d\mathcal{O}_R(I) \subset I \subset \mathcal{O}_L(I)$ coming from the fact that the norm of I is d implies that $\omega \in \mathcal{O}_L(I)$. Since $\mathcal{O}_L(I) \cap \mathcal{O}_R(I) = \mathbb{Z} + I$ it is easily verified that since $n(\omega) = 0 \mod d$, we must have $\omega \in I$ or $\omega \in \overline{I}$. Without loss of generaliy we can assume that $\omega \in I$, and so we have $\mathcal{O}_L(I)\langle \omega, d \rangle \subset I$.

Let d' be the biggest integer such that $\omega/d' \in \mathcal{O}_L(I)$. It is clear that $d' \mid d$. We then set

$$I_1 := \mathcal{O}_L(I)\langle \omega/d', d/d' \rangle$$

where I_1 is a primitive ascending ideal of norm d/d' by Lemma 11. Thus $\mathcal{O}_L(I)\langle \omega, d \rangle = d'I_1 \subset I$.

The ideals I and I_1 are both primitive, contained inside $d'I_1$ and $n(I_1)$ divides $n(I)$ so it is easy to see that I must factor through I_1 and we must have $I \subset I_1$. Hence, we can define

$$I_2 := I_1^{-1} \cdot I$$

By Lemma 11, ω/d defines an optimal embedding of $\mathbb{Z}[\gamma_1]$ into $\mathcal{O}_R(I_1) = \mathcal{O}_L(I_2)$. Since we also had that $\omega/d \in \mathcal{O}_R(I) = \mathcal{O}_R(I_2)$, we conclude that by Lemma 9, I_2 comes from a $\mathbb{Z}[\gamma_1]$-ideal.

Next, we analyse the runtime. Since we are assuming factorization, Heuristic 1 and Heuristic 2, GenericOrderEmbeddingFactorizationAll uses $O(\sqrt{\mathrm{n}(\beta_1)})$ time before returning a potential solution. Enumerating the $\mathbb{Z}[\gamma_1]$-ideals of norm d' can be done efficiently, again by factoring d'. Since there are at most $\mathcal{O}(h(\mathbb{Z}[\gamma_1]))$ isomorphism classes of maximal orders oriented by $\mathbb{Z}[\gamma_1]$, each candidate ideal we end up with has the correct right order with probability

$$O\left(\frac{1}{h(\mathbb{Z}[\gamma_1])}\right) = O\left(\frac{1}{\sqrt{\mathrm{n}(\gamma_1)}}\right),$$

hence we get the expected runtime

$$O\left(\sqrt{\mathrm{n}(\beta_1)\mathrm{n}(\gamma_1)}\right)$$

to find a solution. □

The following corollary is immediate from Proposition 8, but we point it out here for convenience. The first part is generic, and gives the heuristic upper bounded runtime for finding equivalent ideals of given norm, independent of the degree. The second part says that when the orders are special, in the sense that they are both oriented by small quadratic orders, this problem can be solved efficiently, also independent of the degree.

Corollary 6. *Let $\mathcal{O}_1, \mathcal{O}_2 \subset B_{p,\infty}$ be two maximal orders. Then, assuming Heuristic 1, 2, and factorization,* ConnectingIdealWithNorm$_d(\mathcal{O}_1, \mathcal{O}_2)$ *runs in time*

$$O\left(p^{2/3}\right)$$

for any value of d. In the special case that there exists $\gamma_1 \in \mathcal{O}_1$ and $\beta_1 \in \mathcal{O}_2$ with $\mathrm{n}(\beta_1), \mathrm{n}(\gamma_1) \in O(1)$ ConnectingIdealWithSmallNorm$_d(\mathcal{O}_1, \mathcal{O}_2)$ *runs in polynomial time for any d.*

Proof. Immediate from combining Proposition 9, with using Lemma 1 to bound $\mathrm{n}(\gamma_1)$ and $\mathrm{n}(\beta_1)$ by $O(p^{2/3})$ in the generic case, or replacing them with $O(1)$ in the special case.

Appendix B illustrates why the second part of Corollary 6 is particularly interesting, namely because it allows us to compute optimal paths between such orders.

Finally, Algorithm 5 is expected to work in polynomial time for $d = O(p^{2/3})$, when only $\mathcal{O}_1$ contains an element of small norm. However, this expectation completely fails whenever the solution ideal comes from a $\mathbb{Z}[\gamma_1]$-ideal, as the algorithm degenerates into bruteforcing $\mathbb{Z}[\gamma_1]$-ideals. In Appendix C we give another algorithm, which always works in polynomial time in this case, assuming that the third successive minima of $\mathcal{O}_2$ is $O(p^{2/3})$, as one expects for "random" maximal orders.

References

1. Arpin, S., Clements, J., Dartois, P., Eriksen, J.K., Kutas, P., Wesolowski, B.: Finding orientations of supersingular elliptic curves and quaternion orders. arXiv preprint arXiv:2308.11539 (2023)
2. Bencina, B., Kutas, P., Merz, S., Petit, C., Stopar, M., Weitkämper, C.: Improved algorithms for finding fixed-degree isogenies between supersingular elliptic curves. IACR Cryptol. ePrint Arch. p. 1618 (2023), https://eprint.iacr.org/2023/1618
3. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: CSIDH: an efficient post-quantum commutative group action. In: Advances in Cryptology–ASIACRYPT 2018: 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2–6, 2018, Proceedings, Part III 24. pp. 395–427. Springer (2018)
4. Castryck, W., Panny, L., Vercauteren, F.: Rational isogenies from irrational endomorphisms. In: Canteaut, A., Ishai, Y. (eds.) Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12106, pp. 523–548. Springer (2020). https://doi.org/10.1007/978-3-030-45724-2_18, https://doi.org/10.1007/978-3-030-45724-2_18
5. Colò, L., Kohel, D.: Orienting supersingular isogeny graphs. Number-Theoretic Methods in Cryptology 2019 (2019)
6. De Feo, L., Kohel, D., Leroux, A., Petit, C., Wesolowski, B.: SQISign: Compact post-quantum signatures from quaternions and isogenies. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12491, pp. 64–93. Springer (2020). https://doi.org/10.1007/978-3-030-64837-4_3, https://doi.org/10.1007/978-3-030-64837-4_3
7. De Feo, L., Delpech de Saint Guilhem, C., Fouotsa, T.B., Kutas, P., Leroux, A., Petit, C., Silva, J., Wesolowski, B.: Séta: Supersingular encryption from torsion attacks. In: Advances in Cryptology–ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part IV 27. pp. 249–278. Springer (2021)
8. Eisenträger, K., Hallgren, S., Lauter, K.E., Morrison, T., Petit, C.: Supersingular isogeny graphs and endomorphism rings: Reductions and solutions. In: Nielsen, J.B., Rijmen, V. (eds.) Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part III. Lecture Notes in Computer Science, vol. 10822, pp. 329–368. Springer (2018). https://doi.org/10.1007/978-3-319-78372-7_11, https://doi.org/10.1007/978-3-319-78372-7_11
9. Feo, L.D., Fouotsa, T.B., Kutas, P., Leroux, A., Merz, S.P., Panny, L., Wesolowski, B.: SCALLOP: scaling the CSI-FiSh. In: IACR International Conference on Public-Key Cryptography. pp. 345–375. Springer (2023)
10. Kohel, D.: Endomorphism rings of elliptic curves over finite fields. Ph.D. thesis, University of California at Berkeley (1996)

11. Kohel, D., Lauter, K., Petit, C., Tignol, J.P.: On the quaternion-isogeny path problem. LMS Journal of Computation and Mathematics **17**(A), 418–432 (2014)
12. Leroux, A.: An effective lower bound on the number of orientable supersingular elliptic curves. In: SAC 2022-Selected Areas in Cryptography (2022)
13. Leroux, A.: A new isogeny representation and applications to cryptography. In: Advances in Cryptology–ASIACRYPT 2022: 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5–9, 2022, Proceedings, Part II. pp. 3–35. Springer (2022)
14. Love, J., Boneh, D.: Supersingular curves with small noninteger endomorphisms. Open Book Series **4**(1), 7–22 (2020)
15. Onuki, H.: On oriented supersingular elliptic curves. Finite Fields Their Appl. **69**, 101777 (2021). https://doi.org/10.1016/J.FFA.2020.101777, `https://doi.org/10.1016/j.ffa.2020.101777`
16. Page, A., Robert, D.: Introducing Clapoti(s): Evaluating the isogeny class group action in polynomial time. IACR Cryptol. ePrint Arch. p. 1766 (2023), `https://eprint.iacr.org/2023/1766`
17. The Sage Developers: SageMath, the Sage Mathematics Software System (version 9.7) (2022), `https://sagemath.org`
18. Voight, J.: Quaternion Algebras. Springer Graduate Texts in Mathematics series (2018)
19. Wesolowski, B.: Orientations and the supersingular endomorphism ring problem. In: Dunkelman, O., Dziembowski, S. (eds.) Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part III. Lecture Notes in Computer Science, vol. 13277, pp. 345–371. Springer (2022). https://doi.org/10.1007/978-3-031-07082-2_13, `https://doi.org/10.1007/978-3-031-07082-2_13`
20. Wigert, C.S.: Sur l'ordre de grandeur du nombre des diviseurs d'un entier. Almqvist & Wiksell (1907)

A Searching in the Unique Two-Sided Ideal of Norm p.

The main idea behind GenericOrderEmbedding is to exploit a system of equations on the trace pairings $\mod \Delta$ provided by the formula from Proposition 2. In GenericOrderEmbedding, the final solution α is recovered by enumerating all suitable solutions of the trace pairings system until we find the solution.

One might wonder if we could try a different approach to recover the desired solution more efficiently. To simplify the reasoning we restrict hereafter to the special case where $\mathcal{O}$ is a maximal order in $B_{p,\infty}$.

Let α be the solution we are looking for. Using the trace pairings mod p, we get values $t_1, t_2, t_3 \mod p$. Let us take α_0 any solution to the trace pairing system, meaning that $\mathrm{tr}(\alpha_0) = t$, and $\mathrm{tr}(\alpha_0 \beta_i) = t_i \mod p$.

By linearity of the trace, we have that $\alpha_1 = \alpha - \alpha_0$ is an element that lies in the intersection $\mod p$ of the trace pairing kernels.

Since

$$\mathrm{tr}(\alpha_0 \bar{\alpha}_1) \equiv \mathrm{tr}(\alpha \bar{\alpha}_1) \pmod p$$

and

$$\operatorname{tr}(\alpha\bar{\alpha}_1) = \operatorname{tr}((\alpha_0 + \alpha_1)\bar{\alpha}_1) = \operatorname{tr}(\alpha_0\bar{\alpha}_1) + \operatorname{tr}(\alpha_1\bar{\alpha}_1),$$

we have that $\operatorname{tr}(\alpha_1\bar{\alpha_1}) = 2\mathrm{n}(\alpha_1) \equiv 0 \pmod{p}$, hence α_1 is contained in the unique 2-sided ideal of norm p.

More precisely, it can be shown that the kernel of the trace pairing system mod p has always dimension 2. Writing the kernel as a lattice Λ, we get that α_1 must be contained in $\Lambda + p\mathcal{O}$. Thus, we end trying to find to solve the equation $\operatorname{tr}(\alpha_1) = 0$ and $n(\alpha_1 + \alpha_0) = n$ in $\Lambda + p\mathcal{O}$. This yields a new ternary quadratic form, but it is unclear if it is any easier to solve.

However, we can use this idea to modify algorithm 1 in the case of maximal orders to work with any basis (not necessarily reduced), and achieve the same complexity under some heuristics. The idea is that once an element α_0 is found, which has the correct trace pairings modulo p, the element $-\alpha_1$ lying in the unique two-sided ideal of norm p, will heuristically be the vector in the lattice closest to p whenever $\mathrm{n}(\alpha) = \mathrm{n}(\alpha_0 + \alpha_1) < p^{4/3}$. We summarize this in Algorithm 6.

Algorithm 6 OrderEmbeddingCVP$(\mathcal{O}, t, n)$

```
Input: A maximal order $\mathcal{O} \subset B_{p,\infty}$, two integers $t, n \in \mathbb{Z}$ such that there exists an element of
    trace $t$ and norm $n$ in $\mathcal{O}$.
Output: $\perp$ or $\alpha \in \mathcal{O}$ with $n(\alpha) = n$ and $\operatorname{tr}(\alpha) = t$.
Compute any basis $1, \beta_1, \beta_2, \beta_3$ of $\mathcal{O}$.
Compute $D_i = \operatorname{tr}(\beta_i)^2 - 4n(\beta_i)$ for $1 \le i \le 3$, and $D = t^2 - 4n$.
Compute $s_i$ a square root of $DD_i \mod \Delta$.
Compute an element $\alpha_0$ such that $\operatorname{tr}(\alpha) = t$, and $\operatorname{tr}(\alpha\beta_i) = t_i$ for $1 \le i \le 3$
Compute the lattice $\Lambda$, the trace free part of the the unique two-sided $\mathcal{O}$-ideal of norm $p$.
for Enumerate $-\alpha_1 \in \Lambda$, closest to $\alpha_0$ do
    if $n(\alpha_0 + \alpha_1) = n$ then
        Return $\alpha$.
    end if
end for
```

Remark 2. We remark that this idea can also be used to get the same bound for the algorithm from [1]. Recall that this algorithm works by computing an HNF basis $\beta_1, \beta_2, \beta_3, \beta_4$ of the order, i.e.

$$\begin{aligned} \mathcal{O} = \langle e_{00} + e_{01}i + e_{02}j + e_{03}k,& \\ e_{11}i + e_{12}j + e_{13}k,& \\ e_{22}j + e_{23}k,& \\ e_{33}i\rangle_{\mathbb{Z}}& \end{aligned}$$

Then one finds an element $\alpha_0 = t\beta_1 + x_0\beta_2$ by solving for the trace and and norm modulo p. Then, for a solution α, one is looking for $\alpha_1 := \alpha + \alpha_0$ of the form $\alpha_1 = kp\beta_2 + y\beta_3 + z\beta_4$. It is clear that $p\beta_2, \beta_3, \beta_4$ again generates the trace free part of the unique two-sided ideal of norm p, hence, we can again expect α_1 to be the CVP solution to α_0 in this lattice whenever $\mathrm{n}(\alpha) = \mathrm{n}(\alpha_1 - \alpha_0) < p^{4/3}$.

B A Worked Example

We use Algorithm 7 to compute the shortest path in the 2-isogeny graph between E_0 and E_{1728}, where $j(E_i) = i$.

Let $p = 2^{55} \cdot 3 - 1 \equiv 11 \pmod{12}$. We work in the quaternion algebra

$$B_{p,\infty} = \mathbb{Q} + i\mathbb{Q} + j\mathbb{Q} + k\mathbb{Q}$$

where $i^2 = -1$ and $j^2 = -p$. Let $\mathcal{O}_{1728} \cong \mathrm{End}(E_{1728})$ and $\mathcal{O}_0 \cong \mathrm{End}(E_0)$. Explicitly, fix $\mathcal{O}_{1728}$ to be

$$\mathcal{O}_{1728} = \mathbb{Z} + i\mathbb{Z} + \frac{i+j}{2}\mathbb{Z} + \frac{1+k}{2}\mathbb{Z}.$$

We also know that $\mathcal{O}_0$ contains the element $\omega = \frac{1+\sqrt{-3}}{2}$, where $\mathrm{tr}(\omega) = 1$, and $\mathrm{n}(\omega) = 1$. Hence, we look for the smallest $k \in \mathbb{N}$ such that $\mathbb{Z}[2^k\omega]$ embeds into $\mathcal{O}_{1728}$, by running $\mathsf{ConnectingIdealWithSmallNorm}_{2^k}(\mathcal{O}_{1728}, \mathcal{O}_0)$ with for increasing $k \in [1, 2, \ldots]$. This corresponds to running $\mathsf{GenericOrderEmbeddingFactorizationAll}$ with $t = \mathrm{tr}(2^k\omega) = 2^k$ and $n = \mathrm{n}(2^k\omega) = 2^{2k}$. We find that there exists an optimal embedding

$$\iota : \mathbb{Z}[2^k\omega] \hookrightarrow \mathcal{O}_{1728}$$

For $k = 54$ defined by

$$\iota(2^k\omega) = 9007199254740992 + \frac{19924704230006999}{2}i - \frac{23041705}{2}j - 34653096k,$$

and we use this element to find an ideal connecting $\mathcal{O}_{1728}$ and $\mathcal{O}_0$ of norm 2^{54}. Translating this to an isogeny from

$$E_{1728} : y^2 = x^3 + x$$

We find that the point $K \in E_{1728}$ with

$$x(K) = 86739268981076750i + 69276702275648044, \quad i^2 = -1$$

generates an isogeny to E_0 of degree 2^{54}, corresponding to the shortest path between E_0 and E_{1728} in the 2-isogeny graph.

C Another Algorithm for Finding Equivalent Ideals

As mentioned, the problem with Algorithm 5 is that when (most of) the solution ideal comes from a $\mathbb{Z}[\gamma_1]$-ideal, Algorithm 5 may end up brute-forcing through many horizontal ideals if the degree contains many distinct prime factors. In Algorithm 7, we fix this issue. The idea is to compute embeddings for all elements in a basis of $\mathcal{O}_2$. Then we can recover the solution ideal using Lemma 12, which we state below.

Lemma 12. *Let $\mathcal{O} \subseteq B_{p,\infty}$ be a maximal order, and let I be a primitive right $\mathcal{O}$-ideal of norm d coprime to p. Given a basis $1, \gamma_1, \gamma_2, \gamma_3$ of $\mathcal{O}$. Let d_i be the smallest integer such that $d_i\gamma_i \in \mathcal{O}_L(I)$, and let*

$$I_i = \mathcal{O}_L(I)\langle d_1\gamma_1, d_1\rangle.$$

Then

$$I = I_1 \cap I_2 \cap I_3$$

Proof. I is a primitive ideal connecting its left and right order of norm coprime to p, so all the ideals connecting $\mathcal{O}_L(I)$ and $\mathcal{O}_R(I)$ are contained in I. Indeed, let J be an ideal connecting $\mathcal{O}_L(I)$ and $\mathcal{O}_R(J)$, then $I \cdot \overline{J}$ is an $\mathcal{O}_L(I)$ two-sided ideal, and theory (see [18]) tells us that this ideal must be a scalar multiplied by the unique two-sided ideal of norm p in $\mathcal{O}_L(I)$. Since I has norm coprime to p, J can be factored as I times scalars times the unique two-sided ideal and so J is contained in I.

Let $J = I_1 \cap I_2 \cap I_3$.

In the proof of Proposition 8, we proved that we must have $I \subset I_i$ for $i = 1, 2, 3$ and so we have $I \subset J$.

We will now show that we must have $J \subset I$. For that, we are going to use that $\{x \in \mathcal{O}_L(I) \mid x\mathcal{O}_R(I) \subset \mathcal{O}_L(I)\} \subset I$. It is easy to see that this set $\{x \in \mathcal{O}_L(I) \mid x\mathcal{O}_R(I) \subset \mathcal{O}_L(I)\}$ is an ideal whose left order is $\mathcal{O}_L(I)$ and right order is $\mathcal{O}_R(I)$. Thus, it must be contained in I by what we proved earlier.

Let us now take $x \in J$. For each $i = 1, 2, 3$, there must be $\alpha_i, \beta_i \in \mathcal{O}_L(I)$ such that $x = \alpha_i d_i \gamma_i + \beta_i d_i$.

We are going to show that $x\mathcal{O}_R(I) \subset \mathcal{O}_L(I)$. Let us take $y \in \mathcal{O}_R(I)$. Since $1, \gamma_1, \gamma_2, \gamma_3$ is a basis of $\mathcal{O}_R(I)$, we have that $y = y_0 + \sum_{i=1}^{3} y_i\gamma_i$. Thus, $xy = y_0x + \sum_{i=1}^{3} y_i(\alpha_i d_i \gamma_i + \beta_i d_i)\gamma_i$. With $\gamma_i^2 = \mathrm{tr}(\gamma_i)\gamma_i - n(\gamma_i)$ we get

$$xy = y_0x + \sum_{i=1} y_i(\alpha_i n(\gamma_i))d_i + (\beta_i + \alpha_i \mathrm{tr}(\gamma_i))d_i\gamma_i$$

and it is easy to verify that this belongs to $\mathcal{O}_L(I)$.

This proves that $J \subset I$ and this proves the result.

□

We now give the algorithm.

Algorithm 7 $\mathsf{ConnectingIdealWithSmallNorm}_d(\mathcal{O}_1, \mathcal{O}_2)$

```
Input: Two maximal orders $\mathcal{O}_1, \mathcal{O}_2 \subset B_{p,\infty}$, and an integer $d$.
Output: $\perp$ or an ideal $I$ with $\mathcal{O}_L(I) = \mathcal{O}_1$, $O_R(I) \cong \mathcal{O}_2$, and $n(I) = d$.
  Let $1, \gamma_1, \gamma_2, \gamma_3 \in \mathcal{O}_2$ be a Minkowski-reduced basis of $\mathcal{O}_2$.
  Compute $\mathsf{All}_{\omega_1} = \mathsf{GenericOrderEmbeddingFactorizationAll}(\mathcal{O}_1, n(d\gamma_1), t(d\gamma_1))$
  Compute $\mathsf{All}_{\omega_2} = \mathsf{GenericOrderEmbeddingFactorizationAll}(\mathcal{O}_1, n(d\gamma_2), t(d\gamma_2))$
  Compute $\mathsf{All}_{\omega_3} = \mathsf{GenericOrderEmbeddingFactorizationAll}(\mathcal{O}_1, n(d\gamma_3), t(d\gamma_3))$
  for $\omega_1, \omega_2, \omega_3$ in $\mathsf{All}_{\omega_1} \times \mathsf{All}_{\omega_2} \times \mathsf{All}_{\omega_3}$. do
    Set $d_i$ to be the biggest integer s.t. $\omega_i/d_i \in O_1$ for $i = 1, 2, 3$
    Set $I_i := \mathcal{O}_1\langle \omega_i/d_i, d/d_i \rangle$ for $i = 1, 2, 3$
    Set $I := I_1 \cap I_2 \cap I_3$
    if $\mathcal{O}_R(I) \cong \mathcal{O}_2$ then
      return $I$.
    end if
  end for
  return $\perp$
```

Proposition 9. *Assume the existence of a factorization oracle, and that Heuristic 1 holds. Let β_1 be the smallest non-integer in $\mathcal{O}_1$, and let $1, \gamma_1, \gamma_2, \gamma_3$ be a Minkowski-reduced basis of $\mathcal{O}_2$.* $\mathsf{ConnectingIdealWithSmallNorm}_d$ *always returns a solution I if it exists, or $\perp$ if a solution does not exist, and runs in time*

$$O\left(\max\left\{\left\lceil \frac{d\sqrt{\mathrm{n}(\beta_1)\mathrm{n}(\gamma_3)}}{p} \right\rceil, \left\lceil \frac{d\sqrt{\mathrm{n}(\gamma_1)}}{p} \right\rceil \cdot \left\lceil \frac{d\sqrt{\mathrm{n}(\gamma_2)}}{p} \right\rceil \cdot \left\lceil \frac{d\sqrt{\mathrm{n}(\gamma_3)}}{p} \right\rceil \right\}\right).$$

Proof. First, we show the correctness of the algorithm. Assume that a solution I exists. Then I induces an embedding $d\mathcal{O}_2 \cong d\mathcal{O}_R(I) \subset \mathcal{O}_1 = \mathcal{O}_L(I)$. The isomorphism is given by an element α, i.e. $\alpha \mathcal{O}_2 \alpha^{-1} = \mathcal{O}_R(I)$. Setting $\omega_i := \alpha^{-1} d\gamma_i \alpha$ for $i \in \{1, 2, 3\}$, it follows from Lemma 12 that

$$I = \bigcap_{i=1}^{3} \mathcal{O}_1\langle \omega_i/d_i, d/d_i \rangle$$

where d_i are the biggest integers such that $\omega_i/d_i \in \mathcal{O}_1$, thus showing the correctness of the algorithm.

Next, we analyse the runtime. The first potentially dominating term follows directly from running $\mathsf{GenericOrderEmbeddingFactorizationAll}$ on γ_i sequencially, and noting that $\gamma_1 < \gamma_2 < \gamma_3$. However, when β_1 is sufficiently small, the bottleneck of the algorithm becomes iterating over the cartesian product of the solutions. For each γ_i, we bound the number of solutions with Heuristic 2, giving the second dominating term. □

Thus, from Proposition 9, we see that when $\mathrm{n}(\beta_1) = O(1)$, and $\mathrm{n}(\gamma_3) = O(p^{2/3})$ (as one expects for a random maximal order), Algorithm 7 runs in polynomial time for $d < p^{2/3}$.

Generalized Class Group Actions on Oriented Elliptic Curves with Level Structure

Sarah Arpin, Wouter Castryck, Jonathan Komada Eriksen, Gioella Lorenzon and Fréderik Vercauteren

Accepted at WAIFI 2024. Full version to appear online.

Generalized class group actions on oriented elliptic curves with level structure

Sarah Arpin, Wouter Castryck, Jonathan Komada Eriksen,
Gioella Lorenzon, Frederik Vercauteren

Abstract

We study a large family of generalized class groups of imaginary quadratic number fields K and prove that they act freely and (essentially) transitively on the set of O_K-oriented elliptic curves over a field k (assuming this set is non-empty) equipped with appropriate level structure. This extends, in several ways, a recent observation due to Galbraith, Perrin and Voloch for the ray class group. We show that this leads to a reinterpretation of the action of the class group of a suborder $O \subseteq O_K$ on the set of O-oriented elliptic curves, discuss several other examples, and briefly comment on the hardness of the corresponding vectorization problems.

1 Introduction

A current trend in isogeny-based cryptography is to study isogenies that respect certain *level structure*, or additional information about the curves. The interest by cryptographers has two main catalysts. Firstly, the recently established rapid mixing properties of isogeny graphs of supersingular elliptic curves together with a cyclic subgroup of order N (Γ_N^0-level structure) have led to improved security foundations, e.g., of the distributed generation of supersingular elliptic curves with unknown endomorphism ring [Arp22, BCC$^+$23]; see [CL23b] for a generalization of these rapid mixing results to arbitrary level structure. Secondly, Robert's unconditional break of SIDH [Rob23a] has revealed that the problem of finding an isogeny between two elliptic curves with full Γ_N-level structure is dramatically easier than in the case of plain elliptic curves, at least for N smooth and large enough compared to the degree of the isogeny. The security of several recently proposed variants of SIDH [FMP23, BMP23] also reduces to leveled isogeny problems. Some of these can again be broken much more efficiently than in the unleveled case; see [FFP24] for a recent, systematic discussion.

In this paper we find explicit generalized class groups which act on the set of isomorphism classes of elliptic curves with various types of level structure. This work connects the study of isogenies between elliptic curves with level structure to the group action in the oriented framework of Colò–Kohel [CK20] and Onuki [Onu21]. Briefly recall that, for K an imaginary quadratic field, a K-orientation on an elliptic curve E over a field k, say of positive characteristic p, is an embedding $\iota : K \hookrightarrow \operatorname{End}^0(E)$ of K into the endomorphism algebra of E (assuming that such an embedding exists). For an order O of K, such an orientation is a primitive O-orientation if $O = \iota^{-1} \operatorname{End}(E)$. For a fixed order $O \subseteq K$, the set $\mathcal{E}\ell\ell_k(O)$ of primitively O-oriented elliptic curves up to isomorphism naturally comes equipped with a free and (essentially) transitive action of the class group cl_O by isogenies, see Section 2.4 for more details. Isogenies arising from this class group action are called horizontal. For suitable parameters, this is considered a cryptographic group action, underpinning constructions like CRS [Cou06, RS06], CSIDH [CLM$^+$18, CD20] and SCALLOP [FFK$^+$23, CL23a].

Level structures sneak up in the oriented setting as well, although the situation is more diffuse. Recent work [XZQ23] considers the interaction between a particular family of orientations and Γ_N^0-level structure from the perspective of quadratic forms. For f any prime different from p that splits in K, it is well-known that horizontal f-isogenies automatically preserve the two eigenspaces of any generator σ of the primitive order acting on the f-torsion [BF23, FFP24]; this is level structure amounting to the specification of two independent subgroups of order f. However, our starting point is a more interesting recent observation due to Galbraith, Perrin and Voloch [GPV23] in the case of supersingular elliptic curves over $\mathbb{F}_p$, which is the setting of CSIDH: Just as $\mathrm{cl}(\mathbb{Z}[\sqrt{-p}])$ acts on the set of supersingular elliptic curves over $\mathbb{F}_p$ with endomorphism ring $\mathbb{Z}[\sqrt{-p}]$, its ray class group for modulus (N) acts on supersingular elliptic curves over $\mathbb{F}_p$ with full Γ_N-level structure.

Our goal is to analyze to what extent this latter observation is part of a bigger story. Instead of starting from a type of level structure and trying to devise a corresponding class group action, we invert the viewpoint and start from the action of a *generalized class group* and find the correct level structure to put on isomorphism classes of elliptic curves in order to have a group action. In Section 2 we provide an overview of the relevant background on elliptic curves with level structure, orientations, and on generalized class groups. Our main results are discussed in Section 3, where we study a large family of generalized class groups, study their properties, and show that they act freely and (essentially) transitively on oriented elliptic curves with suitable level structure. We discuss several interesting examples, one of which sheds a new light on actions by class groups of non-maximal orders. Finally, in Section 4, we discuss the hardness of the vectorization problem for our generalized class group actions. In [GPV23], the authors observe that although using supersingular elliptic curves over $\mathbb{F}_p$ equipped with level structure provides a larger key space for the usual CSIDH parameters, the security of such an enhanced protocol immediately reduces to the security of the original CSIDH protocol, so this does not provide an advantage. We generalize this observation, and contrast it with class group actions of suborders as in [FFK$^+$23].

2 Background

2.1 Elliptic Curves with Level Structure

In this section, fix an integer $N \geq 2$ and a field k such that $\operatorname{char} k \nmid N$, and let E be an elliptic curve over k. Let $E[N] \cong \mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z}$ denote the N-torsion group of E.

Definition 2.1 (Level structure)**.** *Let* Γ *be a subgroup of* $\mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$*. A level structure on an elliptic curve* E *is a choice of isomorphism* $\Phi : \mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z} \cong E[N]$*, up to pre-multiplication on the right by an element of* Γ*. The triple* (E, Φ, Γ) *is called an elliptic curve with level-*N Γ *structure, and when* Γ *is understood we will drop it from the triple and write* (E, Φ)*.*

Choosing such an isomorphism $\Phi : \mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z} \cong E[N]$ amounts to specifying a basis $P, Q \in E[N]$ and considering it up to base change by matrices from the prescribed group $\Gamma \subseteq \mathrm{GL}_2(\mathbb{Z}/(N))$. An isogeny $\varphi : E_1 \to E_2$ respects the level structures P_1, Q_1 resp. P_2, Q_2 if and only if $\varphi(P_1) = Q_1$, $\varphi(P_2) = Q_2$, modulo the action of Γ. Commonly studied examples are

$$\Gamma_N^0 = \left\{ \begin{pmatrix} * & * \\ 0 & * \end{pmatrix} \right\}, \quad \Gamma_N^{0,0} = \left\{ \begin{pmatrix} * & 0 \\ 0 & * \end{pmatrix} \right\}, \quad \Gamma_N^1 = \left\{ \begin{pmatrix} 1 & * \\ 0 & * \end{pmatrix} \right\}, \quad \Gamma_N = \left\{ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right\}$$

where the level structure corresponds to specifying a cyclic subgroup of order N, two independent subgroups of order N, a point of order N, or a basis of $E[N]$ ("full level structure"), respectively.

2.2 Congruence subgroups and generalized class groups

Our main reference for this and the next section is [Cox13]. For K an imaginary quadratic number field with ring of integers O_K, a *modulus* in K is a non-zero integral ideal $\mathfrak{m} \subseteq O_K$. We denote by I_K the group of non-zero fractional ideals in K, which we recall are lattices $\mathfrak{a} \subseteq K$ such that

$$\{ \alpha \in K \mid \alpha \, \mathfrak{a} \subseteq \mathfrak{a} \} = O_K.$$

Equivalently, a non-zero fractional ideal is any set $\mathfrak{a} \subseteq K$ for which there exists $\alpha \in K \backslash \{0\}$ such that $\alpha \, \mathfrak{a}$ is a non-zero ideal of O_K. Let P_K be the subgroup of non-zero principal fractional ideals, i.e., fractional ideals of the form αO_K with $\alpha \in K \setminus \{0\}$. Then the class group of K is the quotient $\mathrm{cl}_K = I_K / P_K$.

It can be shown that for any choice of modulus $\mathfrak{m}$, every class in cl_K contains an ideal $\mathfrak{a} \subseteq O_K$ that is coprime with $\mathfrak{m}$, i.e., $\mathfrak{a} + \mathfrak{m} = O_K$. Equivalently, if we define $I_K(\mathfrak{m}) \subseteq I_K$ to be the subgroup of non-zero fractional ideals that are coprime with $\mathfrak{m}$ (i.e., have $\mathfrak{p}$-adic valuation 0 for every prime ideal $\mathfrak{p} \mid \mathfrak{m}$) and $P_K(\mathfrak{m}) = P_K \cap I_K(\mathfrak{m})$, then the natural map $I_K(\mathfrak{m})/P_K(\mathfrak{m}) \to \mathrm{cl}_K : [\mathfrak{a}] \mapsto [\mathfrak{a}]$ is an isomorphism.

A *ray* for modulus $\mathfrak{m}$ is a principal fractional ideal of the form

$$(\alpha) \text{ with } \alpha \in K^* \text{ such that } \alpha \equiv 1 \bmod \mathfrak{m}$$

(where the congruence means that $\alpha - 1$ has positive $\mathfrak{p}$-adic valuation for all prime ideals $\mathfrak{p} \mid \mathfrak{m}$). The rays form a subgroup $P_{K,1}(\mathfrak{m}) \subseteq P_K(\mathfrak{m})$ called the *ray group* for modulus $\mathfrak{m}$. Any group H such that

$$P_{K,1}(\mathfrak{m}) \subseteq H \subseteq I_K(\mathfrak{m})$$

is then called a *congruence subgroup* for modulus $\mathfrak{m}$. The corresponding $I_K(\mathfrak{m})/H$ quotient is known as a *generalized class group*; such groups play a crucial role in the study of abelian extensions of K. In the extremal case $H = P_{K,1}(\mathfrak{m})$ one ends up with the ray class group $\mathrm{cl}_{K,1}(\mathfrak{m})$. The ray class group $\mathrm{cl}_{K,1}(\mathfrak{m})$ acts on the set of elliptic curves with endomorphism ring O_K equipped with full level structure, as studied in [GPV23]. Recall that in the intermediate case $H = P_K(\mathfrak{m})$, $I_K(\mathfrak{m})/P_K(\mathfrak{m}) \cong \mathrm{cl}_K$. We will return to this in Section 3, but first we describe generalized class groups of non-maximal orders.

2.3 Class groups of orders

An order $O \subseteq K$ is a free $\mathbb{Z}$-submodule of rank 2 that is also a subring. It can be shown that O_K is the unique maximal order with respect to inclusion, i.e., $O \subseteq O_K$ for any order O. More concretely, $O = \mathbb{Z} + fO_K$ for a unique positive integer f called the *conductor* of O. As in the maximal case one can consider the group I_O of non-zero fractional O-ideals in K, which are lattices $\mathfrak{a} \subseteq K$ such that

$$\{\, \alpha \in K \mid \alpha\, \mathfrak{a} \subseteq \mathfrak{a} \,\} = O.$$

It remains true that every such lattice admits an $\alpha \in K \setminus \{0\}$ such that $\alpha\, \mathfrak{a}$ is a non-zero ideal of O. However, the converse fails: if $O \subsetneq O_K$ then ideals whose norm is not coprime with f may not arise in this way. The non-zero principal fractional O-ideals, i.e., fractional O-ideals of the form αO with $\alpha \in K \setminus \{0\}$, form a subgroup $P_O \subseteq I_O$, and the corresponding quotient $\mathrm{cl}_O = I_O/P_O$ is the class group of O.

Of particular relevance for the current paper is that cl_O can also be realized as a generalized class group of K for modulus $\mathfrak{m} = (f)$.

Theorem 2.2. *Let*

$$P_{K,\mathbb{Z}}((f)) = \{\, (\alpha) \mid \alpha \in K^* \textit{ and } \alpha \equiv g \bmod (f) \textit{ for some } g \in \mathbb{Z} \textit{ coprime with } f \,\}.$$

Then the map

$$\mathrm{cl}_O \to I_K((f))/P_{K,\mathbb{Z}}((f)) : [\mathfrak{a}] \mapsto [\mathfrak{a}\, O_K],$$

where it can be assumed that $[\mathfrak{a}]$ is represented by a fractional O-ideal of norm coprime with (f), is an isomorphism of groups.

Proof. See [Cox13, Prop. 7.22]. □

Remark 2.3. It is also possible to consider the case $O' \subsetneq O$ where the superorder is not maximal, and realize $\mathrm{cl}_{O'}$ as a generalized class group of O. We plan to treat the relative case more carefully in the final version of this work.

The following classical exact sequence provides a foundational framework for working with generalized class groups of orders.

Theorem 2.4 ([Neu99, Theorem I.12.12])**.** *Let K be an imaginary quadratic number field with ring of integers O_K. Let $O \subsetneq O_K$ denote an order of K with conductor f. Then, there is an exact sequence*

$$1 \longrightarrow O_K^\times / O^\times \longrightarrow (O_K/fO_K)^\times/(O/fO_K)^\times \longrightarrow \mathrm{cl}(O) \longrightarrow \mathrm{cl}_K \longrightarrow 1.$$

2.4 Class group actions on sets of elliptic curves

Let k be a field of characteristic $p > 0$. In this section, we describe the class group actions on certain sets of elliptic curves over finite fields. We begin following an approach of Waterhouse [Wat69], who provides such a group action for isomorphism classes of ordinary elliptic curves over finite fields. For ordinary elliptic curves, the class group of the endomorphism ring is used to define a group action on the set of isomorphism classes of curves with isomorphic endomorphism rings. We write $\mathrm{End}(E)$ to denote the ring of all endomorphism rings of E/k, defined over $\overline{k}$. When it arises, we write $\mathrm{End}_k(E)$ to specify the subring of endomorphisms of E defined over k.

Theorem 2.5 ([Wat69, Theorem 4.5])**.** *Let E be an ordinary elliptic curve over a field k of finite characteristic p with endomorphism ring* $\mathrm{End}(E)$*. Then the class group of* $\mathrm{End}(E)$ *acts freely and transitively on the set of elliptic curves over k with endomorphism ring isomorphic to* $\mathrm{End}(E)$*.*

The ideal class group of $\mathrm{End}(E)$ acts on the set of isomorphism classes of ordinary elliptic curves with endomorphism rings isomorphic to the order $\mathrm{End}(E)$ in the following sense:

Definition 2.6. *Let E/k be an elliptic curve over a field k of characteristic p with commutative endomorphism ring* $\mathrm{End}(E)$*. Take an integral ideal I of* $\mathrm{End}(E)$ *with $N(I)$ coprime to p and the conductor of* $\mathrm{End}(E)$*. Define*

$$E[I] := \bigcap_{\alpha \in I} \ker \alpha.$$

As I is a finitely generated $\mathbb{Z}$-module, the set $E[I]$ is a finite group. This finite group defines an isogeny $\varphi_I : E \to E/E[I]$ with kernel $E[I]$. Define

$$I * E := E/E[I].$$

Each ideal class contains an integral ideal representative which is of norm coprime to p and the conductor of $\mathrm{End}(E)$. The principal ideals are generated by a single endomorphism, and so act trivially. In [Wat69, Section 3], Waterhouse establishes that, in the case where E is ordinary, this is a free and transitive group action on the set of elliptic curves with endomorphism ring isomorphic to $\mathrm{End}(E)$. The proof goes through showing that ideals of $\mathrm{End}(E)$ satisfy certain properties qualifying them as *kernel ideals*.

In the supersingular case, the situation is more complicated. If E is supersingular, then $\mathrm{End}(E)$ is a noncommutative ring in a quaternion algebra. In particular, $\mathrm{End}(E)$ does not have a class group of (left or right) ideals. There is a partial remedy and a full remedy. The partial remedy: if $k = \mathbb{F}_p$, then $\mathrm{End}_k(E)$ is isomorphic to an order in the imaginary quadratic field $\mathbb{Q}(\sqrt{-p})$. In this case, one can work with the class group of $\mathrm{End}_k(E)$ and use Definition 2.6, just as in the ordinary case. The group action will be free and transitive (modulo a minor subtlety highlighted in the proof of [Sch87, Thm. 4.5]). However, if $k = \mathbb{F}_{p^n}$ for $n > 1$ or $k = \overline{\mathbb{F}_p}$, we find ourselves in need of additional framework: orientations on supersingular elliptic curves are the full remedy. The remainder of this section deals with the supersingular case.

Let k denote a field of finite characteristic p, and consider a supersingular elliptic curve E/k. By [Sil09], the endomorphism ring $\mathrm{End}(E)$ is isomorphic to a maximal order in the quaternion algebra $B_{p,\infty} := \mathrm{End}(E) \otimes_{\mathbb{Z}} \mathbb{Q}$ ramified precisely at p and ∞. Any non-scalar element $\alpha \in \mathrm{End}(E) \setminus \mathbb{Z}$ generates an imaginary quadratic order. Let K be an imaginary quadratic field in which p does not split. This condition gives the existence of an embedding of K into $B_{p,\infty}$ [Voi21, Prop. 14.6.7].

Definition 2.7. *A K-orientation on an elliptic curve E is an embedding*

$$\iota : K \hookrightarrow \mathrm{End}(E) \otimes_{\mathbb{Z}} \mathbb{Q}.$$

For an order O of K such an embedding is called an O-primitive orientation if $\iota(O) = \iota(K) \cap \mathrm{End}(E)$. The pair (E, ι) is called an O-primitively oriented supersingular elliptic curve. We denote by $\mathcal{E\ell\ell}_k(O)$ the set of primitively O-oriented elliptic curves over k, and denote by $\mathrm{SS}_k(O) \subseteq \mathcal{E\ell\ell}_k(O)$ the subset of primitively O-oriented supersingular elliptic curves.

Remark 2.8. When E is supersingular as above, a K-orientation ι maps into a four-dimensional $\mathbb{Q}$-algebra. In the case where the endomorphism ring of E is commutative, Definition 2.7 can still apply: the map ι defines an isomorphism $O \cong \mathrm{End}(E)$. Note that in both cases we call such a map an O-primitive K-orientation on E, to unify notation.

Definition 2.9 (K-oriented isogeny)**.** *A K-oriented isogeny is an isogeny $\varphi : (E_0, \iota_0) \to (E_1, \iota_\varphi)$ between K-oriented elliptic curves such that $\varphi : E_0 \to E_1$ as an isogeny of elliptic curves and $\iota_\varphi(-) = \frac{1}{\deg \varphi} \varphi \circ \iota_0(-) \circ \widehat{\varphi}$.*

An isomorphism of elliptic curves is likewise a K-oriented isomorphism if it is of degree-1 and satisfies the above properties.

Via the Deuring lifting theorem, the theory of oriented supersingular elliptic curves is closely related to the theory of CM elliptic curves.

Definition 2.10. *There is an extension L' of the ring class field L of O and a prime $\mathfrak{p}$ above p in $O_{L'}$ such that every elliptic curve with CM by O has a representative defined over L' with good reduction at $\mathfrak{p}$.*
Let $\mathcal{E}\ell\ell_{L'}(O)$ denote the set of isomorphism classes of elliptic curves with endomorphism ring isomorphic to O and having good reduction over $\mathfrak{p}$.
Let $\rho : \mathcal{E}\ell\ell_{L'}(O) \to \mathcal{E}\ell\ell_k(O)$ denote the reduction map modulo $\mathfrak{p}$.

Definition 2.11. *Let $(E, \iota) \in \mathcal{E}\ell\ell_k(O)$ and take an ideal $\mathfrak{m}$ of O. Define the (group-theoretic) intersection:*

$$E[\iota(\mathfrak{m})] := \bigcap_{\alpha \in \mathfrak{m}} \ker(\iota(\alpha)).$$

Let $\varphi_{\mathfrak{m}}$ denote a K-oriented isogeny of (E, ι) with kernel $E[\iota(\mathfrak{m})]$. Such an isogeny $\varphi_{\mathfrak{m}} : (E, \iota) \to (E_{\mathfrak{m}}, \iota_{\mathfrak{m}})$ is unique up to K-oriented automorphism on the codomain.
When $\#E[\iota(\mathfrak{m})] = N(\mathfrak{m})$, we define the action of $\mathfrak{m}$ on (E, ι) to be:

$$\mathfrak{m} * (E, \iota) = (E_{\mathfrak{m}}, \iota_{\mathfrak{m}}).$$

Definition 2.11 is precisely the supersingular analogue of Definition 2.6 that we need. The principal ideals (β) of O are generated by endomorphisms of E, and thus act trivially on (E, ι) since $\beta \in O$ commutes with the image of ι in $\mathrm{End}(E)$. The following theorem of Onuki completes the picture by providing the supersingular analogue of Theorem 2.5.

Theorem 2.12 ([Onu21, Theorem 3.4])**.** *When p is not split in the imaginary quadratic field K and p is coprime to the conductor of the imaginary quadratic order O of K, then Definition 2.11 gives a free and transitive action of cl_O on $\rho(\mathcal{E}\ell\ell_k(O))$.*

To understand when this gives a free and transitive action on the set $\mathrm{SS}_k(O)$ of O-primitively oriented supersingular elliptic curves, we need to understand the relationship between the sets $\rho(\mathcal{E}\ell\ell_k(O))$ and $\mathrm{SS}_k(O)$:

Corollary 2.13 ([Sch87, Theorem 4.5])**.** *If p is ramified in the imaginary quadratic field $O \otimes_{\mathbb{Z}} \mathbb{Q}$, then $\mathrm{cl}(O)$ acts freely and transitively on the set of primitively-O oriented supersingular elliptic curves over $\overline{\mathbb{F}_p}$. If p is inert in the imaginary quadratic field $O \otimes_{\mathbb{Z}} \mathbb{Q}$, then $\mathrm{cl}(O)$ has two orbits in the set of primitively-O oriented supersingular elliptic curves over $\overline{\mathbb{F}_p}$.*

Concretely, if p is not inert in the imaginary quadratic field K containing O, then $\mathrm{cl}(O)$ acts freely and transitively on the set of isomorphism classes of primitively O-oriented elliptic curves $\mathcal{E}\ell\ell_k(O)$. If p is split in K, $\mathcal{E}\ell\ell_k(O)$ is a set of ordinary elliptic curves. If p is ramified, $\mathcal{E}\ell\ell_k(O) = \mathrm{SS}_k(O)$ is a set of primitively O-oriented supersingular elliptic curves. If p is inert in K, $\mathrm{cl}(O)$ acts on $\rho(\mathcal{E}\ell\ell_k(O))$, which is again a set of primitively O-oriented supersingular elliptic curves.

3 Generalized class group actions

Let K be an imaginary quadratic number field and let O be an order in K. Let k be a field of characteristic $p > 0$. In this section, if p is inert in K, we fix the orbit $\rho(\mathcal{E}\ell\ell_k(O))$ of $\mathrm{cl}(O)$ and call it $\mathcal{E}\ell\ell_k(O)$ by an abuse of notation.

Let $\mathfrak{m}$ be a modulus in K. Let H be a congruence subgroup for $\mathfrak{m}$ that is contained in $P_K(\mathfrak{m})$. Let

$$\mathrm{cl}_H = I_K(\mathfrak{m})/H$$

be the corresponding generalized class group. Because $H \subseteq P_K(\mathfrak{m})$ the map

$$\mathrm{cl}_H \times \mathcal{E}\ell\ell_k(O_K) \to \mathcal{E}\ell\ell_k(O_K) : ([\mathfrak{a}], E) \mapsto \varphi_{\mathfrak{a}}(E) = E/E[\mathfrak{a}]$$

remains a well-defined group action. However, if $H \subsetneq P_K(\mathfrak{m})$ then this no longer yields a free group action: the class of any ideal $\mathfrak{a} \in P_K(\mathfrak{m}) \setminus H$ is a non-trivial element acting trivially. This creates room for an action of cl_H on elements of $\mathcal{E}\ell\ell_k(O_K)$ equipped with extra data, i.e., with $\mathfrak{m}$-*level structure*, which we now define.

3.1 $\mathfrak{m}$-level structures

Our starting observation is:

Lemma 3.1. *Let O_K be an imaginary quadratic order and let $E \in \mathcal{E}\ell\ell_k(O_K)$. Suppose $\mathfrak{m}$ is an O_K-ideal such that $\#E[\mathfrak{m}] = N(\mathfrak{m})$. Then*

$$E[\mathfrak{m}] \cong O_K/\,\mathfrak{m}$$

as O_K-modules; in particular, they are are also isomorphic as groups.

Proof. By the Deuring lifting theorem, there exists an elliptic curve $\widetilde{E}/\mathbb{C}$ such that $\mathrm{End}(\widetilde{E}) \cong O_K$ and $E[\mathfrak{m}] \cong \widetilde{E}[\mathfrak{m}]$. By [Sil94, Proposition II.1.4], $\widetilde{E}[\mathfrak{m}]$ is a free rank-1 $O_K/\,\mathfrak{m}$-module, so we have the desired isomorphism. □

Remark 3.2. For E supersingular, the condition $\#E[\mathfrak{m}] = N(\mathfrak{m})$ is equivalent to requiring $N(\mathfrak{m})$ coprime to $\mathrm{char}\, k$.

Applying Lemma 3.1 to $\mathfrak{m} = (N)$ for some integer N coprime to $\mathrm{char}\, k$, we recover the well-known fact that

$$E[N] \cong O_K/(N) \cong \mathbb{Z}/(N) \times \mathbb{Z}/(N)$$

as groups, where upon writing $O_K = \mathbb{Z}[\sigma]$ for some generator σ, an instance of the last isomorphism is given by $1 \mapsto (1, 0)$, $\sigma \mapsto (0, 1)$. This motivates the following generalization of Definition 2.1.

Definition 3.3. *Let $\mathfrak{m} \subseteq O_K$ be an ideal coprime to* $\operatorname{char} k$*. Let $\Gamma \subseteq \mathrm{GL}(O_K/\mathfrak{m})$ be a subgroup and let E be an elliptic curve primitively oriented by O_K. A Γ-level structure on E is then a group isomorphism*

$$\Phi : O_K/\mathfrak{m} \to E[\mathfrak{m}]$$

defined up to pre-composition with an element $\gamma \in \Gamma$ and post-composition with a K-oriented automorphism. We denote by Y_Γ the set of primitively O_K-oriented elliptic curves equipped with a Γ-level structure, up to K-oriented isomorphisms. If Γ consists of O_K-module automorphisms, then we denote by $Z_\Gamma \subseteq Y_\Gamma$ the subset for which the level structure is an isomorphism of O_K-modules.

The reason for highlighting the subset Z_Γ will become apparent in Section 3.2.

Remark 3.4. Considering Γ-level structures up to K-oriented automorphisms amounts to identifying (E, Φ) and $(E, \iota(u) \circ \Phi)$ for every $u \in O_K^\times$; here ι denotes the implicit embedding of O_K in $\mathrm{End}(E)$. However, in most cases this can be ignored because it is already taken care of by the Γ-level structure. E.g., this is true if $O_K^\times = \{\pm 1\}$ and Γ is closed under negation.

More concretely, in view of the lemma below, defining a Γ-level structure amounts to specifying a point P of order $a_\mathfrak{m}$ and a point Q of order $b_\mathfrak{m}$ such that $\frac{a_\mathfrak{m}}{b_\mathfrak{m}} P, Q$ is a basis of $E[b_\mathfrak{m}]$, considered up to "base changes" as specified by the subgroup Γ.

Lemma 3.5. *Let $\mathfrak{m}$ be a modulus in K. Then there exist unique $a_\mathfrak{m}, b_\mathfrak{m} \in \mathbb{Z}$ such that* $\operatorname{char} k \nmid b_\mathfrak{m} \mid a_\mathfrak{m}$ *and*

$$E[\mathfrak{m}] \cong \frac{\mathbb{Z}}{(a_\mathfrak{m})} \times \frac{\mathbb{Z}}{(b_\mathfrak{m})}$$

for all $E \in \mathcal{E\ell\ell}_k(O_K)$.

Proof. This is standard: every finite subgroup of E admits such a decomposition, and the independence of E follows because any two such curves are connected by a horizontal isogeny of norm coprime with $\mathfrak{m}$. □

In order to motivate the next sections, let us conclude by restating (a slightly extended version of) the observation made by Galbraith, Perrin, and Voloch [GPV23] in the context of CSIDH. Here one considers $\mathfrak{m} = (N)$ and $\Gamma = \{\mathrm{id}\}$; for simplicity we will just write Y_N, Z_N instead of $Y_{\{\mathrm{id}\}}, Z_{\{\mathrm{id}\}}$. Putting an $\{\mathrm{id}\}$-level structure on a curve $E \in \mathcal{E\ell\ell}_k(O_K)$ just amounts to choosing a basis $P, Q \in E[N]$, i.e., a full level-N structure. Writing $O_K = \mathbb{Z}[\sigma]$, elements of $Z_N \subseteq Y_N$ correspond to bases of the form $P, \sigma(P)$; it is a consequence of Lemma 3.5 that such bases indeed exist.

Theorem 3.6. *Let N be a positive integer coprime to* $\operatorname{char} k$*. Then the ray class group*

$$\mathrm{cl}_{K,1}((N)) = I_K((N))/P_{K,1}((N))$$

acts freely on both Y_N and Z_N; in the latter case, the action is also transitive.

Proof. This is a special case of Theorem 3.8 below. □

3.2 A family of congruence subgroups

The goal of this section (and of this paper) is to embed Theorem 3.6 in a more general story. We concentrate on congruence subgroups of the form

$$P_{K,\Lambda}(\mathfrak{m}) = \{\, (\alpha) \mid \alpha \in K^\times \text{ and } \alpha \equiv \lambda \bmod \mathfrak{m} \text{ for some } \lambda \in \Lambda \text{ coprime to } N(\mathfrak{m}) \,\},$$

where Λ is a multiplicatively closed subset of O_K. This covers the aforementioned congruence subgroups $P_{K,\mathbb{Z}}((f))$ and $P_{K,1}(\mathfrak{m}) = P_{K,\{1\}}(\mathfrak{m})$ as special cases, yet it also introduces several interesting new examples.

Remark 3.7. Notice that Λ and $\pm\Lambda$ or more generally $O_K^\times \Lambda$ define the same congruence subgroup, as one can always change the generator α of a principal ideal accordingly. Thus it would make sense to impose $O_K^\times \subseteq \Lambda$. However, we refrain from doing this, in order to keep covering standard notation such as $P_{K,\mathbb{Z}}((f))$; also the exact sequence from Proposition 3.10 is affected by this, see Remark 3.12.

Now, to such a congruence subgroup $P_{K,\Lambda}(\mathfrak{m})$ we can naturally associate the subgroup

$$\Gamma_{K,\Lambda}(\mathfrak{m}) = \{\, \mu_\alpha \mid (\alpha) \in P_{K,\Lambda}(\mathfrak{m}) \,\} = \{\, \mu_\lambda \mid \lambda \in O_K^\times \Lambda \,\} \subseteq \mathrm{GL}(O_K/\,\mathfrak{m})$$

where μ_α refers to the action of multiplication by α on $O_K/\,\mathfrak{m}$. By definition of $P_{K,\Lambda}(\mathfrak{m})$, this is a multiplicative subset of the finite group $\mathrm{GL}(O_K/\,\mathfrak{m})$, hence indeed a subgroup. Note that the μ_α's are O_K-module automorphisms, so both $Y_{\Gamma_{K,\Lambda}(\mathfrak{m})}$ and $Z_{\Gamma_{K,\Lambda}(\mathfrak{m})}$ are well-defined.

Theorem 3.8. *Let $H = P_{K,\Lambda}(\mathfrak{m})$ be as above. Then*

$$[\mathfrak{a}] \star (E, \Phi) = (\phi_{\mathfrak{a}}(E), \phi_{\mathfrak{a}} \circ \Phi) \tag{1}$$

is a well-defined free action of cl_H on $Z_{\Gamma_{K,\Lambda}(\mathfrak{m})}$. Moreover, this action is transitive. If $\Lambda \subseteq O_K^\times \mathbb{Z}$ then this extends to a free action of cl_H on $Y_{\Gamma_{K,\Lambda}(\mathfrak{m})}$.

Proof. Since $\deg \varphi_{\mathfrak{a}} = N(\mathfrak{a})$ is assumed coprime with $\mathfrak{m}$, it follows readily that the right-hand side of (1) is an element of $Y_{\Gamma_{K,\Lambda}(\mathfrak{m})}$. Using that $\varphi_{\mathfrak{a}}$ is K-oriented, we also see that it concerns an element of $Z_{\Gamma_{K,\Lambda}(\mathfrak{m})}$ as soon as (E, Φ) is.

Now assume $(E, \Phi) \in Z_{\Gamma_{K,\Lambda}(\mathfrak{m})}$ and let $\mathfrak{a} = (\alpha)$ be the principal ideal generated by some $\alpha \in O_K$. Then

$$\varphi_{(\alpha)} \circ \Phi = \Phi \circ \mu_\alpha \tag{2}$$

because Φ is an isomorphism of O_K-modules. It follows that Φ and $\varphi_{(\alpha)} \circ \Phi$ define the same $\Gamma_{K,\Lambda}(\mathfrak{m})$-level structure on E if and only if $(\alpha) \in P_{K,\Lambda}(\mathfrak{m})$. But this implies that the action is well-defined and free. As for the transitivity, it suffices to argue that if

$$\Phi_1, \Phi_2 : O_K/\,\mathfrak{m} \to E[\mathfrak{m}]$$

are two isomorphisms as O_K-modules, then there exists $\alpha \in O_K$ such that $\Phi_2 = \varphi_{(\alpha)} \circ \Phi_1$. This is evident from the fact that we are dealing with free rank-1 modules over $O_K / \mathfrak{m}$.

Finally, we need to show that if $\Lambda \subseteq O_K^\times \mathbb{Z}$ then we still have have a well-defined and free action. By ignoring post-compositions with K-oriented automorphisms, we can in fact assume $\Lambda \subseteq \mathbb{Z}$. For this we need to show that

$$\varphi_{(\alpha)} \circ \Phi = \Phi \circ \mu_{\alpha'}$$

for some $(\alpha') \in P_{K,\Lambda}(\mathfrak{m})$ if and only if $(\alpha) \in P_{K,\Lambda}(\mathfrak{m})$. Since we are working modulo $\mathfrak{m}$, this amounts to saying that

$$\varphi_{(\alpha)} \circ \Phi = \Phi \circ \mu_\lambda = [\lambda] \circ \Phi$$

for some $\lambda \in \Lambda$ if and only if $(\alpha) \in P_{K,\Lambda}(\mathfrak{m})$; the last equality follows because Φ is a group homomorphism. If $(\alpha) \in P_{K,\Lambda}(\mathfrak{m})$ then the existence of such a λ is clear. On the other hand, if such a λ exists then from [Sil09, Cor. III.4.11] it follows that $\alpha \equiv \lambda \bmod \mathfrak{m}$, as wanted. (Note that, in the above reasoning, we have used that we can ignore units, in view of Remark 3.4.) □

In general, the action of the generalized class group cl_H on $Y_{\Gamma_{K,\Lambda}(\mathfrak{m})}$ is far from transitive. E.g., recall from Theorem 3.6 that the ray class group acts freely on

$$Y_{\Gamma_N} = \{ (E, P, Q) \mid E \in \mathcal{E}\ell\ell_k(O_K),\ P, Q \text{ basis of } E[N] \},$$

but when writing $O_K = \mathbb{Z}[\sigma]$, it is easy to see that if P happens to be an eigenvector of σ, this can never be "undone" by acting with a ray class. There are two natural ways to make the action more transitive:

- Restricting to a subset of $Y_{\Gamma_{K,\Lambda}(\mathfrak{m})}$; this is exactly what we did above when studying $Z_{\Gamma_{K,\Lambda}}$, which seems to be the most natural option.
- Further identifying elements of $Y_{\Gamma_{K,\Lambda}}$ by working with a bigger group $\Gamma \supseteq \Gamma_{K,\Lambda}$.

We now analyse the action of cl_H on a set defined by $\Gamma \supseteq \Gamma_{K,\Lambda}(\mathfrak{m})$. First, note that we are free to chose any such set, as the following lemma shows.

Lemma 3.9. *Assume $\Lambda \subseteq \mathbb{Z}$, let $H = P_{K,\Lambda}(\mathfrak{m})$ and consider the free action of cl_H on $Y_{\Gamma_{K,\Lambda}}$ from above. Then this descends to a well-defined action of cl_H on Y_Γ for any $\Gamma \supseteq \Gamma_{K,\Lambda}(\mathfrak{m})$.*

Proof. The set Y_Γ consists of equivalence classes of elements of $Y_{\Gamma_{K,\Lambda}}$. Thus, we only need to show that if $(E, \Phi) \sim (E, \Phi')$, i.e. $\Phi' = \Phi \circ T$ for some $T \in \Gamma$, then $(\phi_\mathfrak{a}(E), \phi_\mathfrak{a} \circ \Phi) \sim (\phi_\mathfrak{a}(E), \phi_\mathfrak{a} \circ \Phi')$ for some $[\mathfrak{a}] \in \mathrm{cl}_H$. But this is clearly true, since we still have $\phi_\mathfrak{a} \circ \Phi' = \phi_\mathfrak{a} \circ \Phi \circ T$. □

The class group cl_H surjects onto cl_K, and as the action of cl_K is well understood, we aim to study the action of the kernel of this surjection. To do this, we start by slightly generalising the well-known exact sequence from Theorem 2.4:

Proposition 3.10. *Let $\mathfrak{m}, \Lambda$ and K be as above. Let $H = P_{K,\Lambda}(\mathfrak{m})$. Then Λ defines a subgroup of $(O_K/\mathfrak{m})^\times$, defined as $\Delta := \phi(\Lambda) \cap (O_K/\mathfrak{m})^\times$, where ϕ denotes the natural surjection from O_K to $O_K/\mathfrak{m}$. Then, there is an exact sequence*

$$1 \to O_K^\times/(O_K^\times \cap (\Lambda + \mathfrak{m})) \to (O_K/\mathfrak{m})^\times/\Delta \to \mathrm{cl}_H \to \mathrm{cl}_K \to 1$$

Proof. The proof closely follows the proof from Cox [Cox13, Theorem 7.24], which proves the special case $\Lambda = \mathbb{Z}$, $\mathfrak{m} = (f)$ from Theorem 2.4.

We prove this from the right to left. The surjection $\pi : \mathrm{cl}_H \to \mathrm{cl}(O_K)$ is obtained from the natural map sending $[\mathfrak{a}] \in I_K(\mathfrak{m})/P_{K,\Lambda}(\mathfrak{m})$ to the class of $\mathfrak{a}$ in $I_K(\mathfrak{m})/P_K(\mathfrak{m}) \cong \mathrm{cl}(O_K)$. The kernel is therefore exactly $P_K(\mathfrak{m})/P_{K,\Lambda}(\mathfrak{m})$. Next, we show that there is a surjection

$$(O_K/\mathfrak{m})^\times/\Delta \to P_K(\mathfrak{m})/P_{K,\Lambda}(\mathfrak{m})$$

obtained by sending $[[\alpha]] \in (O_K/\mathfrak{m})^\times/\Delta$ to αO_K (we will use the notation $[\gamma]$ for elements of $(O_K/\mathfrak{m})^\times$, and $[[\gamma]]$ for elements of $(O_K/\mathfrak{m})^\times/\Delta$). The ideal αO_K is clearly in $P_K(\mathfrak{m})$. Further, let $[\alpha] = [\beta][\delta]$, for some $\delta \in \Lambda$, i.e. α and β are in the same class of $(O_K/\mathfrak{m})^\times/\Delta$. Then, unraveling the definitions, there exists some $u \in O_K$ such that $u\alpha \equiv u\beta\delta \equiv 1 \pmod{\mathfrak{m}}$. Further, we can choose some $\delta' \in \Lambda$ such that $[\delta'] \equiv [\delta]^{-1}$. Thus, we have that

$$\alpha O_K \cdot u\beta\delta O_K = \beta O_K \cdot u\alpha\delta^{-1} O_K,$$

which shows that the map is a well defined group homomorphism, since $u\beta\delta O_K \in P_{K,1}(\mathfrak{m}) \subseteq P_{K,\Lambda}(\mathfrak{m})$, and $u\alpha\delta^{-1}O_K \in P_{K,\Lambda}(\mathfrak{m})$.

Next, we show that the map is surjective. Let $\gamma O_K \in P_K(\mathfrak{m})$. Obviously, if $\gamma \in O_K$, then $[[\gamma]]$ maps to γO_K. In general, γ can be written as $\gamma_1\gamma_2^{-1}$ for $\gamma_1, \gamma_2 \in O_K$ (since O_K is maximal), which both are coprime to $\mathfrak{m}$. Thus, we have that the class $[[\gamma_1][\gamma_2]^{-1}]$ maps to γO_K, proving that the map is surjective.

Finally, assume $[\alpha] \in (O_K/\mathfrak{m})^\times$ satisfies $\alpha O_K \in P_{K,\Lambda}(\mathfrak{m})$. Thus, we have that $\alpha O_K = \beta\gamma^{-1}O_K$, for some β, γ satisfying $[\beta'][\gamma]^{-1} \in \Delta$, and that $\alpha = \mu\beta\gamma^{-1}$ for some $\mu \in O_K^\times$. This in turn means that $[\alpha] = [\mu][\beta][\gamma]^{-1}$, and since $[\beta][\gamma]^{-1} \in \Delta$, we see that $[[\alpha]] = [[\mu]]$, i.e. $[[\alpha]]$ is in the image of the obvious homomorphism $O_K^\times \to (O_K/\mathfrak{m})^\times/\Delta$, whose kernel is in turn exactly $O_K^\times \cap (\Lambda + \mathfrak{m})$. □

Remark 3.11. We can compare Proposition 3.10 with both Theorem 2.4 and the formula for computing the size of the ray class group from [Coh12, Theorem 3.2.4]: Specialising to $\Lambda = \mathbb{Z}$ and $\mathfrak{m} = (f)$, and writing $O := \mathbb{Z} + fO_K$, we immidiately see that

$$O_K^\times \cap (\Lambda + \mathfrak{m}) = O_K^\times \cap (\mathbb{Z} + fO_K) = O^\times,$$

and similarly, $\Delta = \phi(\mathbb{Z}) \cap (O_K/fO_K)^\times = (\mathbb{Z}+fO_K/fO_K)^\times = (O/fO_K)^\times$, recovering the exact sequence from Theorem 2.4.

The size of the ray class group can be computed from the exact sequence when $\Lambda = \{1\}$, in which case one finds that $\Delta = \{[1]\}$, and thus

$$\#\,\mathrm{cl}_H = h(O_K)\frac{\#(O_K/\,\mathfrak{m})^\times}{\frac{O_K^\times}{O_K^\times \cap (1+\mathfrak{m})}} = h(O_K)\frac{\#(O_K/\,\mathfrak{m})^\times}{[O_K^\times : O_{K,\mathfrak{m}}^\times]}$$

where $O_{K,\mathfrak{m}}^\times$ is the group of units congruent to 1 mod $\mathfrak{m}$.

Remark 3.12. Recall from Remark 3.7 that switching from Λ to $O_K^\times\Lambda$ does not affect the congruence subgroup $P_{K,\Lambda}(\mathfrak{m})$, and therefore it does not change the generalized class group either. Also, it does not affect the subgroup $\Gamma_{K,\Lambda}(\mathfrak{m})$. However, it is interesting to observe that it can slightly reorganize the terms in the exact sequence from Proposition 3.10. Indeed, switching from Λ to $O_K^\times\Lambda$ has the effect of folding the exact sequence, which is of the form

$$1 \to G_1 \to G_2 \xrightarrow{f} G_3 \to G_4 \to 1, \qquad \text{into} \qquad 1 \to \frac{G_2}{\ker f} \to G_3 \to G_4 \to 1.$$

Since we know that cl_K acts freely on the set of primitively O_K-oriented curves, we use the surjection $\pi : \mathrm{cl}_H \to \mathrm{cl}(O_K)$ from the exact sequence above, and study the action of $\ker\pi$. By Proposition 3.10, The elements of $\ker\pi$ are principal ideals, which can be identified by elements of $O_K/\,\mathfrak{m}$ up to multiplication by Δ and $O_K^\times$. In particular, they are endomorphisms, leaving the curve fixed, and acting on the different Γ-level structures, which in turn can be identified (by definition) with left cosets of $\Gamma \subset \mathrm{GL}(O_K/\,\mathfrak{m})$, up to K-oriented isomorphisms. This action is fairly easy to describe explicitly, as the following lemma shows.

Corollary 3.13. *Let $K, \Lambda, \mathfrak{m}$ and the map π be as before. Let $\Gamma \supseteq \Gamma_{K,\Lambda}(\mathfrak{m})$. Then, $\ker\pi$ acts on the set*

$$X_\Gamma := \{M\Gamma \mid M \in \mathrm{GL}(O_K/\,\mathfrak{m})\}/\sim$$

where $\sim$ is the equivalence relation obtained by identifying cosets up to left multiplication by μ_u for $u \in O_K^\times$.

Proof. As we have seen, $\ker\pi$ can be identified with elements $[\alpha] \in (O_K/\,\mathfrak{m})^\times$, up to multiplication by Λ and $O_K^\times$. We show that the natural action of sending the left coset $M\Gamma$ to $\mu_\alpha M\Gamma$ is well defined. This is clearly a well defined action by $(O_K/\,\mathfrak{m})^\times$, so it suffices to show that multiplication by elements of Λ and $O_K^\times$ act trivially. Since $\lambda \in \Lambda \subseteq \mathbb{Z}$, its clear that

$$\mu_\lambda M\Gamma = M\mu_\lambda\Gamma = M\Gamma,$$

and further, for $u \in O_K^\times$, μ_u acts trivially by definition of X_Γ. □

3.3 Suborder class group actions

As one of our main examples, let us concentrate on the case where $\Lambda = \mathbb{Z}$ and $\mathfrak{m} = (f)$ for some prime number f different from $\operatorname{char} k$. Pick $\sigma \in O_K$ such that $O_K = \mathbb{Z}[\sigma]$. Write $H = P_{K,\mathbb{Z}}((f))$ and observe that

$$\Gamma := \Gamma_{K,\mathbb{Z}}((f)) \subseteq \mathrm{GL}(O_K/(f))$$

is just the group of multiplications μ_λ by an integer λ that is not divisible by f. Thus we have

$$Y_\Gamma = \{\, (E, P, Q) \mid E \in \mathcal{E\ell\ell}_k(O_K),\ P, Q \text{ basis of } E[f]\}/\sim$$

where $(E, P, Q) \sim (E, \lambda P, \lambda Q)$ for any scalar $\lambda \in (\mathbb{Z}/f\,\mathbb{Z})^\times$. The action of cl_H on Y_Γ is not transitive. Recall that there are two approaches towards turning this into a "more transitive" action:

- Instead of Y_Γ, we can act on Z_Γ, i.e., we can require that the isomorphism

  $$\Phi : O_K/(f) \to E[f]$$

  is an isomorphism of O_K-modules. This amounts to picking a basis of $E[f]$ of the form $P, \sigma(P)$. Note that it suffices to specify P in this case, and because of the scaling we are in fact specifying a cyclic subgroup $C \subseteq E$ of order f. However, since $P, \sigma(P)$ must be a basis, the subgroups we thus obtain are those that are *not* eigenspaces of σ acting on $E[f]$. In other words, we can identify

  $$Z_\Gamma = \{\, (E, C) \mid E \in \mathcal{E\ell\ell}_k(O_K),\ C \subseteq E \text{ kernel of descending } f\text{-isogeny}\}. \qquad (3)$$

  Thanks to Theorem 3.8 we know that cl_H acts freely and transitively on this set.

- Alternatively, we can apply the idea of making the action of cl_H on Y_Γ more transitive by enlarging Γ. In this case it is natural to consider

  $$\Gamma_N^0 = \left\{ \begin{pmatrix} * & * \\ 0 & * \end{pmatrix} \right\} \supseteq \Gamma,$$

  where the last inclusion makes sense upon identifying $\mathrm{GL}(O_K/(f))$ with $(\mathbb{Z}/f\,\mathbb{Z})^2$. Thus by Lemma 3.9 we also have a natural action of cl_H on

  $$Y_N^0 = Y_{\Gamma_N^0} = \{\, (E, C) \mid E \in \mathcal{E\ell\ell}_k(O_K),\ C \subseteq E \text{ cyclic subgroup of order } f\,\}.$$

  However, unless f is inert, this action is neither free nor transitive: any eigenspace of σ acting on $E[f]$ is fixed by every element of cl_H. To turn this into a free and transitive action one has to discard the eigenspaces; as such one again arrives at Z_Γ.

Now let $O = \mathbb{Z} + fO_K$ be the order of conductor f and recall from Theorem 2.2 that the natural map

$$\mathrm{cl}_O \to \mathrm{cl}_H : [\mathfrak{a}] \mapsto [\mathfrak{a}\, O_K] \tag{4}$$

is an isomorphism. In fact, more generally, it is easy to check that the exact sequence from Proposition 3.10 fits in an isomorphism of exact sequences

$$\begin{tikzcd}
1 \arrow[r] & \frac{O_K^\times}{O^\times} \arrow[r] \arrow[d] & \frac{(O_K/fO_K)^\times}{(O/fO_K)^\times} \arrow[r] \arrow[d] & \mathrm{cl}_O \arrow[r] \arrow[d] & \mathrm{cl}_K \arrow[r] \arrow[d] & 1 \\
1 \arrow[r] & \frac{O_K^\times}{O_K^\times \cap (\Lambda + fO_K)} \arrow[r] & \frac{(O_K/fO_K)^\times}{\Delta} \arrow[r] & \mathrm{cl}_H \arrow[r] & \mathrm{cl}_K \arrow[r] & 1
\end{tikzcd}$$

where the vertical maps are the natural maps, and where the sequence on top is the well-known exact sequence from Theorem 2.4.

Now recall from Section 2.4 that we have a free and transitive action of cl_O on $\mathcal{E}\ell\ell_k(O)$. On the other hand, as we have just discussed, there is also a free and transitive action of cl_H on Z_Γ. Finally, we have the isomorphism (4) connecting cl_O to cl_H, as well as a natural bijection

$$Z_\Gamma \to \mathcal{E}\ell\ell_k(O) : (E, C) \mapsto \pi(E, C) := E/C.$$

It can be argued that all these maps are compatible with each other:

Lemma 3.14. *For every ideal class $[\mathfrak{a}] \in \mathrm{cl}_O$ we have*

$$[\mathfrak{a}] \star \pi(E, C) = \pi([\mathfrak{a}\, O_K] \star (E, C)),$$

where the left action is that of cl_O on $\mathcal{E}\ell\ell_k(O)$, while the right action is that of cl_H on Z_Γ.

Proof. Write $E_1 = \pi(E, C)$ and let $E_1' = \varphi_\mathfrak{a}(E_1)$. Then $E \in \mathcal{E}\ell\ell_k(O_K)$ lies above E_1 via an ascending isogeny φ with kernel $E_1[f, f\sigma]$. Likewise, there is an elliptic curve $E' \in \mathcal{E}\ell\ell_k(O_K)$ above E_1', which is the codomain of an ascending isogeny φ' with kernel $E_1'[f, f\sigma]$. It is easy to check that we obtain a commuting diagram

$$\begin{tikzcd}
E \arrow[r, "\varphi_{\mathfrak{a}\, O_K}"] & E' \\
E_1 \arrow[u, "\varphi"] \arrow[r, "\varphi_\mathfrak{a}"] & E_1' \arrow[u, "\varphi'"']
\end{tikzcd}$$

showing that $[\mathfrak{a}\, O_K] \star E = E'$, an equality which refers to the action of cl_K on $\mathcal{E}\ell\ell_k(O_K)$; see also the proof of [Sut13, Lem. 6]. It is then immediate that $C = \ker \hat{\varphi}$ is mapped via $\varphi_{\mathfrak{a}\, O_K}$ to $C' := \ker \hat{\varphi}'$, from which the statement follows. □

Remark 3.15. From the commutativity of the above diagram it follows that

$$\hat{\varphi}' \circ \varphi_{\mathfrak{a}\, O_K} \circ \varphi = \hat{\varphi}' \circ \varphi' \circ \varphi_\mathfrak{a} = [f] \circ \varphi_\mathfrak{a},$$

showing that this is the horizontal isogeny corresponding to the ideal $(f) \cdot \mathfrak{a}$. Since the natural map $\mathrm{cl}_O \to \mathrm{cl}_K$ is not injective, one can also wonder what ideal we end up with when first choosing an isogeny $\psi : E \to E'$ and considering the horizontal isogeny $\hat{\varphi}' \circ \psi \circ \varphi$. One particular case is where $E = E'$, as in the case of SCALLOP. In this case the ideals corresponding to $\hat{\varphi}' \circ \varphi$ have a particularly nice interpretation: they correspond to O-ideals of norm f^2, of the form $\mathfrak{a}_{\alpha,\beta} = (f^2, f(\alpha + \beta\sigma))$, for some $\alpha + \beta\sigma \in O_K$ such that $N_{K/\mathbb{Q}}(\alpha + \beta\sigma) \not\equiv 0 \bmod f$. Indeed, there is in this case a free and transitive action of $\ker(\mathrm{cl}_O \to \mathrm{cl}_K)$ on the set of children of E, namely the set of all $E/C \in \mathcal{E}\ell\ell_k(O)$ over cyclic f-subgroups C of E that are not eigenspaces of σ acting on $E[f]$, which corresponds to the free and transitive action of cl_O on Z_Γ as in Theorem 3.8. It can be proven that ideal classes $[\mathfrak{a}_{\alpha,\beta}]$ in cl_O only depend on $(\alpha : \beta) \in \mathbb{P}^1(\mathbb{F}_f)$ with $N_{K/\mathbb{Q}}(\alpha+\beta\sigma) \not\equiv 0 \bmod f$ and that they constitute all of $\ker(\mathrm{cl}_O \to \mathrm{cl}_K)$ (see also [BLS12, Lemma 3.2]). Once fixed a subgroup $C = \ker\varphi$, the action of such classes is explicitly given by: $[\mathfrak{a}_{\alpha,\beta}] \star \pi(E, C) = \pi(E, (\alpha + \beta\iota(\bar{\sigma}))(C))]$, where $\bar{\sigma}$ is the complex conjugate of σ. This follows from Lemma 3.14 and the fact that $[\mathfrak{a}_{\alpha,\beta}] = [(N_{K/\mathbb{Q}}(\alpha + \beta\sigma), \alpha + \beta\bar{\sigma})]$ with the latter representative being coprime to f.

3.4 Further examples

Cyclic torsion. Assume that $O_K/\mathfrak{m}$ is cyclic, i.e., $b_\mathfrak{m} = 1$ in Lemma 3.5. Then we can observe

- that every $\alpha \in O_K$ is congruent to an integer mod $\mathfrak{m}$; in particular it can be assumed without loss of generality that $\Lambda \subseteq \mathbb{Z}$,
- every group isomorphism $\Phi : O_K/\mathfrak{m} \to E[\mathfrak{m}]$ is necessarily an isomorphism of O_K-modules, i.e., $Y_\Gamma = Z_\Gamma$ for any $\Gamma \subseteq \mathrm{GL}(O_K/\mathfrak{m})$.

As a more concrete example, take $\Lambda = \{1\}$ and let f be a prime number that splits in $O_K = \mathbb{Z}[\sigma]$. Then $fO_K = \mathfrak{m} \cdot \overline{\mathfrak{m}}$ for some prime ideal $\mathfrak{m} \subseteq O_K$. By Theorem 3.8 the ray class group $I_K(\mathfrak{m})/P_{K,1}(\mathfrak{m})$ acts freely and transitively on the set $Z_{\Gamma_{K,1}(\mathfrak{m})}$ of O_K-oriented elliptic curves E/k equipped with an eigenvector of σ acting on $E[f]$. For instance, if $K = \mathbb{Q}(\sqrt{-p})$, f is odd and $p \equiv 1 \bmod f$ as in CSIDH, then we can take

$$\mathfrak{m} = (f, \sqrt{-p} - 1)$$

and this consists of supersingular elliptic curves over $\mathbb{F}_p$ with a distinguished $\mathbb{F}_p$-rational point of order f (up to negation).

Scaling by *n*-th powers. Let O_K be an order and $\mathfrak{m} = (f)$ an ideal such that $f \equiv 1 \pmod 4$ is prime in O_K. Let $\Lambda = \mathbb{Z}^2 := \{\alpha^2 \mid \alpha \in \mathbb{Z}\}$, and assume further that $O_K^\times = \{\pm 1\}$, such that $O_K^\times \subset \Lambda + f\,\mathbb{Z}$ (here we use $f \equiv 1 \pmod 4$). Observe that $H := P_{K,\Lambda}(\mathfrak{m})$ is thus the set of principal ideals generated by elements that are equivalent to integers that are squares $\mod \mathfrak{m}$, and then it follows from 3.10, that $\#\,\mathrm{cl}_H = 2(f+1)h(O_K)$. As in the situation in Subsection 2.4, we get the set $Z_{\Gamma_{K,\Lambda}(\mathfrak{m})}$ consists of elements of the form $(E, P, \sigma(P))$, where $P \in E[f]$, and where we recognise $(E, P, \sigma(P)) \sim (E, Q, \sigma(Q))$ if and only if $P = [\lambda]Q$ for some $\lambda \in \mathbb{Z}$ that is a square $\mathrm{mod} f$. Thus, the situation here is a fine-grained version the action of $\mathrm{cl}(\mathbb{Z}+fO_K)$ on $(E, \langle P\rangle)$ from Subsection 2.4: The slightly larger class group acts on the set that can be recognized as curves, together with one of two points of order f for each subgroup of order f, and where the two points differ by multiplication by a non-square.

This example easily generalizes to $\Lambda = \mathbb{Z}^n$ for any $n \mid f-1$, such that $O_K^\times \subset \Lambda + f\,\mathbb{Z}$.

The full class group. If $\Lambda = O_K$, then $P_{K,\Lambda}(\mathfrak{m}) = P_K(\mathfrak{m})$ is the group of all fractional principal ideals coprime to $\mathfrak{m}$, and we end up with the standard action of cl_K on $\mathcal{E}\ell\ell_k(O_K)$, which indeed naturally coincides with Z_Γ where $\Gamma = \Gamma_{K,O_K}(\mathfrak{m})$. Note that in general we do not have a well-defined action of cl_K on the larger set Y_Γ; indeed the condition $\Lambda \subseteq O_K^\times\,\mathbb{Z}$ from Theorem 3.8 is violated. Nevertheless it makes sense to study Y_Γ as a set; e.g., when $\mathfrak{m} = (N)$ then it parametrizes O_K-oriented elliptic curves E together with a basis of $E[N]$, where two bases are identified if and only if they can be transformed into one another via an endomorphism in $\iota(O_K)$.

4 Security reductions and non-reductions

Although we do not have cryptographic applications in mind, it is natural to extend the central question of [GPV23] to our generalized setting: given

$$(E_1, \Phi_1),\ (E_2, \Phi_2)\ \in\ Z_{\Gamma_{K,\Lambda}(\mathfrak{m})},$$

how hard is it to find a generalized ideal class

$$[\mathfrak{a}] \in I_K(\mathfrak{m})/P_{K,\Lambda}(\mathfrak{m})$$

such that $[\mathfrak{a}](E_1, \Phi_1) = (E_2, \Phi_2)$? This problem is known as the vectorization problem [Cou06], which quantum computers can solve in sub-exponential time $L_h(1/2)$, with h denoting the size of the generalized class group.

Unsurprisingly, the main conclusion from [GPV23, Alg. 2] also applies here: despite the generalized ideal class group being larger, there is an immediate reduction to the vectorization problem for cl_K, potentially at the cost of a discrete logarithm computation (which may be hard classically, but succumbs to Shor's algorithm quantumly). Indeed, after finding an ideal $\mathfrak{a} \in I_K(\mathfrak{m})$ such that $\varphi_\mathfrak{a} : E_1 \to E_2$, one can find $\alpha \in O_K$ such that $(\alpha)\,\mathfrak{a}$ moreover maps Φ_1 to Φ_2, via the computation of Weil pairings and discrete logarithms.

Remark 4.1. It is worth contrasting this with actions by class groups of suborders $O \subseteq O_K$. From Section 3.3 we know that the action of cl_O on $\mathcal{E}\ell\ell_k(O)$ is a generalized class group action in disguise. However, it would be wrong to apply the previous discussion and conclude that the corresponding vectorization problem reduces to that of cl_K acting on $\mathcal{E}\ell\ell_k(O_K)$ at the cost of discrete logarithm computations. There is again a reduction, but it proceeds by walking to $\mathcal{E}\ell\ell_k(O_K)$ via isogenies; in other words, the “disguise” is crucial for security. An extreme case is SCALLOP [FFK⁺23, CL23a], where cl_K is the trivial group, but here the isogenies are of very large prime degree, hence infeasible to compute.

Cases where vectorization becomes easier In view of the attacks on SIDH, the extra level structure may in fact make the vectorization problem much easier. E.g., in the case of the ray class group for scalar modulus $\mathfrak{m} = (N)$, an attacker has access to a basis P_1, Q_1 of $E_1[N]$ along with their images under the secret isogeny $\varphi_{\mathfrak{a}} : E_1 \to E_2$. Assuming we are given a bound on $\deg \varphi_{\mathfrak{a}} = N(\mathfrak{a})$ and assuming that N is large enough and smooth, we can recover $\deg \varphi_{\mathfrak{a}}$ and run the algorithm from [Rob23b] to solve the vectorization problem in classical polynomial time. Another interesting case is the generalized class group action from Section 3.3, where we have a free and transitive action on oriented elliptic curves together with the kernel of a descending f-isogeny, see (3). Thus, here one is given access to such a kernel $C_1 \subseteq E_1[f]$ along with its image $C_2 \subseteq E_2[f]$. But then one also knows that $\sigma(C_1)$ is connected to $\sigma(C_2)$ via the same unknown scalar: the vectorization problem becomes an instance of the M-SIDH problem [FMP23], which can again be broken in overstretched cases. Moreover, if f splits then we also know that the action preserves the eigenspaces of σ acting on the f-torsion; as such we have access to *four* subgroups together with their images, and we can reduce to the case of SIDH via [FP22, Lem. 1].

Supergroups of $P_K(\mathfrak{m})$ Finally, let us drop the overall assumption made at the beginning of Section 3, namely that $H \subseteq P_K(\mathfrak{m})$: what if, conversely, our congruence subgroup H is a supergroup of $P_K(\mathfrak{m})$? In this case the generalized class group $\mathrm{cl}_H = I_K(\mathfrak{m})/H$ naturally acts on subsets

$$\{\, [\mathfrak{h}]E \mid \mathfrak{h} \in H \,\} \subseteq \mathcal{E}\ell\ell_k(O_K) \tag{5}$$

of oriented elliptic curves that can be connected via an ideal in H. In theory, this gives a reduction from the vectorization problem for cl_K to that of the smaller group cl_H: first find the class connecting $\{\, [\mathfrak{h}]E_1 \mid \mathfrak{h} \in H \,\}$ and $\{\, [\mathfrak{h}]E_2 \mid \mathfrak{h} \in H \,\}$ and then solve a vectorization problem for $H/P_K(\mathfrak{m})$. If this were possible, then this could be converted into a Pohlig–Hellman type reduction for class group actions. But unfortunately (or fortunately), it is unclear how to work with the sets $\{\, [\mathfrak{h}]E \mid \mathfrak{h} \in H \,\}$; e.g., when merely working with a representant, one lacks tools for equality testing (which amounts to deciding whether or not two O_K-oriented elliptic curves E_1, E_2 are connected via an ideal in H).

References

[Arp22] Sarah Arpin. *Supersingular Elliptic Curve Isogeny Graphs.* PhD thesis, University of Colorado Boulder, 2022.

[BCC$^+$23] Andrea Basso, Giulio Codogni, Deirdre Connolly, Luca De Feo, Tako Boris Fouotsa, Guido Maria Lido, Travis Morrison, Lorenz Panny, Sikhar Patranabis, and Benjamin Wesolowski. Supersingular curves you can trust. In *EUROCRYPT (2)*, volume 14005 of *Lecture Notes in Computer Science*, pages 405–437. Springer, 2023.

[BF23] Andrea Basso and Tako Boris Fouotsa. New SIDH countermeasures for a more efficient key exchange. In *ASIACRYPT (8)*, volume 14445 of *Lecture Notes in Computer Science*, pages 208–233. Springer, 2023.

[BLS12] Reinier Bröker, Kristin Lauter, and Andrew V. Sutherland. Modular polynomials via isogeny volcanoes. *Mathematics of Computation*, 81(278):1201–1231, 2012.

[BMP23] Andrea Basso, Luciano Maino, and Giacomo Pope. Festa: Fast encryption from supersingular torsion attacks. Springer-Verlag, 2023.

[CD20] Wouter Castryck and Thomas Decru. CSIDH on the surface. In *PQCrypto*, volume 12100 of *Lecture Notes in Computer Science*, pages 111–129. Springer, 2020.

[CK20] Leonardo Colò and David Kohel. Orienting supersingular isogeny graphs. *J. Math. Cryptol.*, 14(1):414–437, 2020.

[CL23a] Mingjie Chen and Antonin Leroux. SCALLOP-HD: group action from 2-dimensional isogenies. *IACR Cryptol. ePrint Arch.*, page 1488, 2023.

[CL23b] Giulio Codogni and Guido Lido. Spectral theory of isogeny graphs, 2023.

[CLM$^+$18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: an efficient post-quantum commutative group action. In *Advances in cryptology—ASIACRYPT 2018. Part III*, volume 11274 of *Lecture Notes in Comput. Sci.*, pages 395–427. Springer, Cham, 2018.

[Coh12] Henri Cohen. *Advanced topics in computational number theory*, volume 193. Springer Science & Business Media, 2012.

[Cou06] Jean Marc Couveignes. Hard homogeneous spaces. *IACR Cryptol. ePrint Arch.*, page 291, 2006.

[Cox13] D.A. Cox. *Primes of the Form $x^2 + ny^2$: Fermat, Class Field Theory, and Complex Multiplication*. Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts. Wiley, 2013.

[FFK$^+$23] Luca De Feo, Tako Boris Fouotsa, Péter Kutas, Antonin Leroux, Simon-Philipp Merz, Lorenz Panny, and Benjamin Wesolowski. SCALLOP: scaling the CSI-FiSh. In *Public-Key Cryptography - PKC 2023 Part I*, volume 13940 of *Lecture Notes in Computer Science*, pages 345–375. Springer, 2023.

[FFP24] Luca De Feo, Tako Boris Fouotsa, and Lorenz Panny. Isogeny problems with level structure. In *EUROCRYPT 2024 (to appear)*. Springer-Verlag, 2024.

[FMP23] Tako Boris Fouotsa, Tomoki Moriya, and Christophe Petit. M-SIDH and MD-SIDH: countering SIDH attacks by masking information. In *EUROCRYPT (5)*, volume 14008 of *Lecture Notes in Computer Science*, pages 282–309. Springer, 2023.

[FP22] Tako Boris Fouotsa and Christophe Petit. A new adaptive attack on SIDH. In *Topics on Cryptology – CT-RSA*, volume 13161 of *Lecture Notes in Computer Science*, pages 322–344. Springer, 2022.

[GPV23] Steven D. Galbraith, Derek Perrin, and José Felipe Voloch. CSIDH with level structure. *IACR Cryptol. ePrint Arch.*, page 1726, 2023.

[Neu99] Jürgen Neukirch. *Algebraic number theory*, volume 322 of *Grundlehren der mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1999. Translated from the 1992 German original and with a note by Norbert Schappacher, With a foreword by G. Harder.

[Onu21] Hiroshi Onuki. On oriented supersingular elliptic curves. *Finite Fields Their Appl.*, 69:101777, 2021.

[Rob23a] Damien Robert. Breaking SIDH in polynomial time. In *EUROCRYPT (5)*, volume 14008 of *Lecture Notes in Computer Science*, pages 472–503. Springer, 2023.

[Rob23b] Damien Robert. Breaking SIDH in polynomial time. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 472–503. Springer, 2023.

[RS06] Alexander Rostovtsev and Anton Stolbunov. Public-key cryptosystem based on isogenies. Cryptology ePrint Archive, Paper 2006/145, 2006. https://eprint.iacr.org/2006/145.

[Sch87] René Schoof. Nonsingular plane cubic curves over finite fields. *J. Combin. Theory Ser. A*, 46(2):183–211, 1987.

[Sil94] Joseph H. Silverman. *Advanced topics in the arithmetic of elliptic curves*, volume 151. Springer, 1994.

[Sil09] Joseph H. Silverman. *The arithmetic of elliptic curves*, volume 106 of *Graduate Texts in Mathematics*. Springer, second edition, 2009.

[Sut13] Andrew V. Sutherland. Isogeny volcanoes. In *ANTS-X*, volume 1 of *Open Book Series*, pages 507–530. MSP, 2013.

[Voi21] John Voight. *Quaternion algebras*, volume 288 of *Graduate Texts in Mathematics*. Springer, Cham, [2021] ©2021.

[Wat69] William C. Waterhouse. Abelian varieties over finite fields. *Ann. Sci. Ecole Norm. Sup.*, 2:521–560, 1969.

[XZQ23] Guanju Xiao, Zijian Zhou, and Longjiang Qu. Oriented supersingular elliptic curves and eichler orders, 2023.

PEARL-SCALLOP: Parameter Extension Applicable in Real Life for SCALLOP

Bill Allombert, Márton Tot Bagi, Jean-françois Biasse, Jonathan Komada Eriksen, Péter Kutas, Chris Leonardi, Aurel Page and Renate Scheidler

In submission. Full version to appear online.

PEARL-SCALLOP: Parameter Extension Applicable in Real Life for SCALLOP

Bill Allombert, Márton Tot Bagi, Jean-François Biasse, Jonathan Komada Eriksen,
Péter Kutas, Chris Leonardi, Aurel Page, Renate Scheidler

Abstract. Cryptographic group actions are useful tools of designing various cryptographic primitives reminiscent of discrete logarithm-based primitives that are post-quantum secure. One important technical issue is to provide commutative cryptographic group actions where the structure of the underlying group can be computed efficiently as that enables several advanced primitives such as advanced signature schemes (threshold signatures, group signatures, ring signatures etc.). One such instantiation is SCALLOP which uses supersingular elliptic curves oriented by non-maximal orders of prime conductors in fields of small class number (e.g., prime conductor suborders of $\mathbb{Z}[i]$).

We present PEARL-SCALLOP a variant of SCALLOP changing several parameter and design choices. The goal of modifying the original framework is to improve both on efficiency and security. We propose using maximal orders with larger class numbers and non-maximal orders of non-smooth (but not prime) conductors. This way parameter generation can be accomplished efficiently for larger parameter sets and the order of the class group is not smooth as that might be desirable for security purposes. As an important subroutine we propose a practical algorithm for generating oriented supersingular elliptic curves.

To demonstrate our improvements, we provide a proof-of-concept implementation which instantiates PEARL-SCALLOP at a higher security-level than any other SCALLOP variant has been instantiated at, and show that our improvements give timings which are more than an order of magnitude faster than any previous implementation.

1 Introduction

Isogeny-based cryptography dates back to Couveignes' seminal work [25] where he introduced the concept of hard homogeneous spaces which are today often referred to as cryptographic group action [1] as a quantum-resistant alternative of the usual Diffie-Hellman key exchange [31]. Later Rostovtsev and Stolbunov [48] rediscovered Couveignes' ideas and the resulting scheme is dubbed the CRS key exchange.

The CRS key exchange utilizes the group action of certain class groups of imaginary quadratic orders on the set of ordinary elliptic curves. The security of the scheme relies on the hardness of inverting the group action. Unfortunately, constructions based on ordinary curves are rather slow. A breakthrough result in this topic was CSIDH [18] where the key idea is to switch from ordinary elliptic curves to supersingular ones defined over $\mathbb{F}_p$. On this set, there is a natural group action of the class group of $\mathbb{Z}[\sqrt{-p}]$ that can be utilized to build a key exchange.

De Feo and Galbraith built a signature schemes using CSIDH together with the Fiat-Shamir with aborts technique called SeaSign [28]. The difficulty (and hence inefficiency) of CSIDH-based signatures is that for cryptographically sized parameters it is hard to compute the structure of the class group. For CSIDH-512 Beullens, Kleinjung and Vercauteren computed the structure of the group using a record-breaking computation which they then applied to build the signature scheme CSi-FiSh [10]. The framework of CSi-FiSh can then be applied to build threshold signatures [29], ring signatures [9], group signatures [8] and many more primitives.

Unfortunately, due to [46] and [14] it is unclear whether CSIDH-512 (and thus CSi-FiSh) achieve NIST level I security hence it is important to have instantiations with larger parameters. Even though CSIDH easily generalizes to higher security levels, CSi-FiSh would require class group computations that are out of reach for current algorithms and computational resources. SeaSign does scale for larger parameter sets but is highly impractical.

The notion of an orientation by an arbitrary imaginary quadratic order was formally introduced by Colò and Kohel, when they introduced the OSIDH protocol [24]. Recently De Feo, Fouotsa, Kutas, Leroux, Merz, Panny and Wesolowski proposed SCALLOP [36] which is a different cryptographic group action, building on this notion of an orientation. The key idea of SCALLOP is to use a non-maximal order of large prime conductor in a quadratic number field of small class number. The class number of these orders can be calculated easily using the class number formula. Then computing the structure of the class group reduces to computing certain discrete logarithms in said class groups. By carefully generating parameters this allows one to implement signature schemes for security levels comparable to CSIDH-512 and CSIDH-1024 without the need to use particularly large resources (i.e., pre-computations can be carried out on a laptop). However, from the original construction it is a bit unclear whether SCALLOP can be instantiated for security levels comparable to CSIDH-2048 and CSIDH-4096, and SCALLOP is significantly slower than CSIDH.

SCALLOP-HD is a variant of SCALLOP that uses higher dimensional tools developed in [27] to provide a polynomial-time parameter generation. The reason for this is as follows. In SCALLOP the natural generator of the order has non-smooth degree. Since one needs to evaluate this endomorphism one has to find a way of representing it in a compact way. In SCALLOP this is done by writing it as a linear combination of 1 and a smooth degree endomorphism which is a non-trivial task in the parameter generation phase. SCALLOP-HD bypasses this by representing the isogeny using higher dimensional techniques just as in SQISignHD. One important trick is that it only needs to use isogenies in dimension 2 as opposed to SQISignHD which uses isogenies in dimension 4 (or 8). On the other hand, SCALLOP-HD does not yet have a public implementation, so it is not clear how it compares to the original SCALLOP scheme.

1.1 Our contributions

We make different design choices than in SCALLOP for security and efficiency purposes. First we will work with a maximal order that has a class number that is large but still efficiently computable (i.e., have a discriminant of roughly 256 bits) and use conductors to define the non-maximal order $\mathfrak{O}$ that are not smooth but also not prime (but the product of a few, large primes). Choosing such a conductor defeats all the attacks already considered in SCALLOP hence does not seem to pose a security threat. Furthermore, using a large maximal order on top ensures that the class number will not be smooth, hence is potentially more secure against hidden shift attacks.

In the original SCALLOP the conductor f is chosen to be prime and in a way that $f \pm 1$ is smooth in order to utilize the Pohlig-Hellmann algorithm for discrete logarithm computations. This makes the class group computation easy but becomes hard to achieve for larger security levels. By switching to the product of large primes we can reduce the class group computations to mid-size discrete logarithm computations in finite fields that can be computed efficiently in practice.

They main benefit of this construction is that group action evaluation is significantly faster than in SCALLOP and SCALLOP-HD. The extra flexibility in our parameter generation allows one to represent the orientation with an endomorphism whose degree is a power of 2. This way we do not require higher dimensional isogeny representations and do not need to evaluate higher dimensional isogenies for translating the orientation (i.e., we replace the chain of $(2, 2)$-isogenies with 2-isogenies). Furthermore, we can use odd degree isogenies in the group action evaluation thus we do not run into the expensive issue of SCALLOP where the norm of ideal to be evaluated is not coprime to the norm of the endomorphism that represents the orientation.

As a subroutine we design a more efficient algorithm for generating oriented curves together with the orientation. In theory this can be accomplished in polynomial time using the maximal order to elliptic curve algorithm from [33] or its more practical variant [35]. However, for larger parameter sets generating a supersingular elliptic curve with prescribed endomorphism ring is computationally very expensive.

Putting all of these pieces together we propose a new SCALLOP variant, PEARL-SCALLOP, that we instantiate for the security level comparable to CSIDH-512, CSIDH-1024 and CSIDH-2048. When defining security levels we will always compare to versions of CSIDH (as the quantum bit security of Kuperberg's algorithm instantiated for class groups is debated).

1.2 Technical overview

We give a more detailed analysis of the technical ideas of the paper and make comparison between SCALLOP, SCALLOP-HD and PEARL-SCALLOP.

It is known [45] that one can instantiate class group actions with any orientation. However, there are three important requirements when designing efficient signature schemes and more advanced primitives:

- Security: Disclosing the orientation should not reveal too much information about the endomorphism ring of the curve
- Efficient representation: The orientation should have an efficient representation that allows one to evaluate the class group action
- Efficiently computable class group structure

CSIDH satisfies the first two criteria but its class group structure (ore even class number) can't be efficiently computed for larger security levels as CSIDH-512 was already a record class group computation. The idea of SCALLOP is to use non-maximal orders of large conductor. In SCALLOP and SCALLOP-HD the maximal quadratic order has small class number (in the proposed parameters it has class number 1). It would be natural to use a non-maximal order of smooth conductor as then the orientation would have an efficient representation and an oriented curve can be computed by a single smooth-degree isogeny evaluation. Unfortunately such a construction is insecure because of the following. Let ι be the endomorphism of the curve oriented by the maximal order. Then the orientation of the non-maximal order corresponds to the endomorphism $\tau = \phi \circ \iota\hat{\phi}$. Now we can evaluate τ on any point of powersmooth degree and then basically recover ϕ using techniques developed in [30].

In order to avoid such an attack one can use orders of non-smooth conductors. However, then two problems arise. First one has to be able to represent this orientation. The way this is achieved in SCALLOP is that one represents the orientation with a smooth generator. A smooth generator always exists but there are several challenges in making this construction practical. First finding a smooth generator takes usually subexponential time and then the smoothness bounds are highly impractical (which effects the runtime of the group action evaluation significantly). Thus in SCALLOP one somehow finds the smooth generator first and then tries to find a suitable f. Since we need $f - 1$ to be smooth, this puts some restrictions on the particular structure of this generator which provide the biggest slowdown of SCALLOP.

In SCALLOP-HD the trick is to use higher dimensional isogeny representation. Now one choose any prime conductor f where $f \pm 1$ is smooth. This is easy as one can just simply choose a prime of the form $2^k 3^l h - 1$ where h is a small cofactor. This construction allows for polynomial-time class group computations hence scales well for any security level. The minor drawback here is that one needs to use higher dimensional isogenies for translating the orientation and the class group has smooth degree (which might or might not be a problem for certain applications or improved hidden shift attacks).

The idea of PEARL-SCALLOP is to use a maximal order of larger discriminant on top whose class group is still efficiently computable. We use conductors that are not primes but are not smooth, they are a product of a few primes, depending on the security levels. We revisit the idea of representing orientations by smooth generators. The advantage now comes from the fact that we will be looking for the conductor and the maximal order together. Let us look for a and d such that $d + a^2$ is a fixed power of 2 (this is the norm of $a + \sqrt{-d}$) and the coefficient of $\sqrt{-d}$ in a small power of $a + \sqrt{-d}$ is the product of a few primes. This has two benefits. First we can represent our orientation with an isogeny whose degree is a power of 2. Second we can reduce the class group computation to computing the class group of the maximal order and discrete logarithm computations in moderate sized finite fields. This approach is going to be faster than SCALLOP-HD but does not scale in polynomial time (as eventually the discrete logarithm computations will become too expensive). One extra benefit is that the class group will not have smooth degree as the maximal order has non-smooth discriminant and the prime factors of the conductor are not special primes.

One difficulty in using suborders of maximal orders for large class numbers is that one has to be able to generate an oriented curve. This can be done in polynomial time but is extremely costly in practice even for 1000-bit primes. Our new idea to make this construction more practical is as follows. Assume that we want to generate an $\mathfrak{O} = \mathbb{Z}[\omega]$-oriented curve. First let us take a smooth positive integer f such that $\mathbb{Z}[f\omega]$ embeds into the curve $E : y^2 = x^3 + x$. This involves the solution of a relatively simple Diophantine equation and it is known that every order embeds into $End(E)$ if it embeds into $B_{p,\infty}$ and its discriminant is bigger than $O^*(p^2)$. Then using the efficient representation of $End(E)$ one only has to compute an ascending f-isogeny to arrive at a curve oriented by $\mathfrak{O}$. The efficiency gain comes from the fact that previous algorithms computed the orientation on the quaternion side first and then did a full maximal order to elliptic curve algorithm. Note that this algorithm could be interesting on its own or for further variants of SCALLOP.

The paper is structured as follows. In Section 2 we recall some necessary mathematical preliminaries and the high-level idea and design choices of SCALLOP [36]. In Section 3 we present our new framework and propose algorithms to generate parameters. In Section 4 we discuss the concrete instantiation, implementation challenges and our novel algorithm for generating an oriented curve (Algorithm 1).

2 Preliminaries

In this section, we recall the main theoretical concepts needed for understanding SCALLOP, and the class group computation.

2.1 Supersingular elliptic curves and orientations

We begin with a brief review of the required background material on elliptic curves and their orientations. For details, we refer the reader to [36] and the sources cited therein.

Let $p \geq 5$ be a prime and $\overline{\mathbb{F}}_p$ an algebraically closed field of characteristic p. For any elliptic curve $E/\overline{\mathbb{F}}_p$ and any non-negative integer n, we denote by $E[n]$ the group of n-torsion points on E, i.e. the kernel of the multiplication-by-n map on E. Throughout, we will only consider *supersingular* elliptic curves, i.e. curves $E/\overline{\mathbb{F}}_p$ for which $E[p]$ is trivial. Since every supersingular elliptic curve is isomorphic to a curve defined over $\mathbb{F}_{p^2}$, we may assume that E is given by a short Weierstrass equation

$$E : y^2 = x^3 + Ax + B$$

with $A, B \in \mathbb{F}_{p^2}$.

For any isogeny $\phi : E \to E'$, let $\hat{\phi}$ denote its dual and $\deg(\phi)$ its degree. All isogenies herein are assumed to be separable; in particular, $p \nmid \deg(\phi)$ and $\deg(\phi) = \# \ker(\phi)$ is the cardinality of the kernel of ϕ. The only exception is the p-power Frobenius isogeny $\pi : E \to E^p$ defined via $\pi((x, y)) = (x^p, y^p)$, where E^p is given by $y^2 = x^3 + A^p x + B^p$.

Let $\mathrm{End}(E)$ denote the endomorphism ring of E and $\mathrm{End}^0(E) = \mathrm{End}(E) \otimes_{\mathbb{Z}} \mathbb{Q}$ the associated endomorphism algebra. Then $\mathrm{End}^0(E) \cong B_{p,\infty}$, the rational quaternion algebra ramified at p and ∞, and $\mathrm{End}(E)$ is isomomorphic to a maximal order of $B_{p,\infty}$.

Let K be an imaginary quadratic field such that p does not split in K. Then K embeds into $B_{p,\infty}$. A K-*orientation* of E is a (necessarily injective) ring homomorphism $\iota : K \to \mathrm{End}^0(E)$. If $\phi : E \to E'$ is an isogeny, then ϕ induces a K-orientation ι' of E' defined via

$$\iota'(\beta) = \frac{1}{\deg(\phi)} \, \phi \circ \iota(\beta) \circ \hat{\phi} \quad \text{for all } \beta \in K.$$

If there exists an order $\mathfrak{O} \subset K$ (which is unique in this case) such that $\iota(\mathfrak{O}) = \mathrm{End}(E) \cap \iota(K)$, then ι is said to be an $\mathfrak{O}$-*orientation.*[1] The curve E is then said to be $\mathfrak{O}$-*orientable* and the pair (E, ι) is referred to as an $\mathfrak{O}$-*oriented* elliptic curve.

Note that every $\mathfrak{O}$-orientation ι of E gives rise to an orientation on $E^p = \pi(E)$, since $\mathrm{End}(E) \cong \mathrm{End}(E^p)$. The set $S_{\mathfrak{O}}(p)$ of $\mathfrak{O}$-oriented elliptic curves up to isomorphism and Frobenius conjugacy is non-empty if and only if p does not divide the conductor of $\mathfrak{O}$. If in addition p splits in K, then $\# S_{\mathfrak{O}}(p) = h(\mathfrak{O})$, the class number of $\mathfrak{O}$. In this case, the class group $\mathrm{Cl}(\mathfrak{O})$ acts freely and transitively on $S_{\mathfrak{O}}(p)$ as follows. For an $\mathfrak{O}$-oriented curve (E, ι_E) and an $\mathfrak{O}$-ideal $\mathfrak{a}$ coprime to the conductor of $\mathfrak{O}$, put $E[\mathfrak{a}] = \bigcap_{\alpha \in \mathfrak{a}} \ker(\iota_E(\alpha))$ and let $\varphi_{\mathfrak{a}}^E : E \to E/E[\mathfrak{a}]$ be the isogeny with kernel $E[\mathfrak{a}]$, of degree $N(\mathfrak{a})$ where $N(\mathfrak{a}) = [\mathfrak{O} : \mathfrak{a}]$ is the norm of $\mathfrak{a}$. Then $\mathfrak{a} \star (E, \iota_E) = (E_{\mathfrak{a}}, \iota_{\mathfrak{a}}) \in S_{\mathfrak{O}}(p)$, where

$$E_{\mathfrak{a}} = E/E[\mathfrak{a}], \qquad \iota_{\mathfrak{a}}(\beta) = \frac{1}{N(\mathfrak{a})} \, \varphi_{\mathfrak{a}}^E \circ \iota_E(\beta) \circ \hat{\varphi}_{\mathfrak{a}}^E \quad \text{for all } \beta \in K.$$

[1] In some sources, $\mathfrak{O}$-orientations are referred to as *primitive* $\mathfrak{O}$-orientations (with $\mathfrak{O}$-orientations without this attribute only requiring $\iota(\mathfrak{O}) \subseteq \mathrm{End}(E) \cap \iota(K)$), or as *optimal embeddings*.

Since principal ideals act trivially on $S_{\mathfrak{O}}(p)$, this extends to an action $\star : \mathrm{Cl}(\mathfrak{O}) \times S_{\mathfrak{O}}(p) \to S_{\mathfrak{O}}(p)$. In practice, $\varphi_{\mathfrak{a}}^{E}$ will always be given as a product of low degree isogenies of coprime degrees, corresponding to a factorization of $\mathfrak{a}$ into powers of prime ideals in $\mathfrak{O} = \mathbb{Z}[\omega]$. Specifically, if $\mathfrak{a} = \mathfrak{b}\mathfrak{c}$, where $\mathfrak{b}$, $\mathfrak{c}$ are $\mathfrak{O}$-ideals whose norms relatively prime to each other and to the conductor of $\mathfrak{O}$, then $E_{\mathfrak{b}}[\mathfrak{c}] = \varphi_{\mathfrak{b}}^{E}(E[\mathfrak{c}])$ and $\varphi_{\mathfrak{a}}^{E} = \varphi_{\mathfrak{c}}^{E_{\mathfrak{b}}} \circ \varphi_{\mathfrak{b}}^{E}$. If $\mathfrak{c}$ is primitive, i.e. not divisible by any rational integers other than ±1, and given by a $\mathbb{Z}$-basis $\{c, u+\omega\}$ with $c = N(\mathfrak{c})$, then $E[\mathfrak{c}]$ is a cyclic group, computable as $E[\mathfrak{c}] = E[c] \cap \ker(\iota_E(\omega) + [u])$, where $[u]$ is the multiplication-by-u map on E.

2.2 SCALLOP

In this subsection we describe the main mechanism and design choices of SCALLOP [36]. As explained in the previous section, every $\mathfrak{O}$-orientation yields an action of $\mathrm{Cl}(\mathfrak{O})$ on the set of $\mathfrak{O}$-oriented supersingular elliptic curves. The aim of SCALLOP was to find an orientation with the following properties:

1. The class number of $\mathfrak{O}$ is easy to compute
2. The relation lattice of $\mathrm{Cl}(\mathfrak{O})$ is easier to compute than in CSIDH (i.e., for the same security levels)
3. Computing the endomorphism ring of an $\mathfrak{O}$-oriented curve (even when the orientation is provided) is hard (i.e., there does not exists a quantum polynomial-time algorithm for computing the endomorphism ring)

In order to satisfy the first condition, SCALLOP uses non-maximal orders of quadratic fields with small class number. Assuming the factorization of the conductor is known, class numbers of such orders are easy to compute using the formula

$$h(\mathfrak{O}) = \frac{h(\mathfrak{O}_K) f}{[\mathfrak{O}_K^* : \mathfrak{O}^*]} \prod_{q \mid f} \left(1 - \left(\frac{d_K}{q}\right) \frac{1}{q}\right);$$

see [26, Theorem 7.24]. Here, $\mathfrak{O}_K$ is the maximal order of the field of fractions K of $\mathfrak{O}$, d_K is its discriminant, $\mathfrak{O}_K^*$ and $\mathfrak{O}^*$ are the respective unit groups of $\mathfrak{O}_K$ and $\mathfrak{O}$, f is the conductor of $\mathfrak{O}$, and the product runs over all primes dividing f.

The third condition is somewhat trickier to satisfy in this case. As a particular example, let $\mathfrak{O}$ be an order in $\mathbb{Z}[i]$ of smooth conductor f. Given an $\mathfrak{O}$-orientable elliptic curve E, one can recover a degree f endomorphism of E in the following fashion. Let $\mathfrak{O} = \mathbb{Z}[\omega]$ and $\sigma = \iota(\omega)$ be the natural generator of $\mathrm{End}(E)$, which we decompose as $\sigma = \phi \circ [i] \circ \hat{\phi}$. Here, $\phi : E_0 \to E$ is an isogeny of degree f, the curve E_0 has j-invariant 1728 and is $\mathbb{Z}[i]$-orientable, and $[i]$ is the endomorphism corresponding to i. Then σ/f is the $\mathfrak{O}$-orientation of E induced by ϕ. We know how to evaluate σ on any point on E. Then choosing an appropriate scalar d, one can ensure that $\sigma + d$ has degree B where B is powersmooth. In this way, we obtain a generator $\tau = \sigma + d \in \mathrm{End}(E)$ as a composition of low-degree isogenies.

Then one can compute $\ker(\tau - d) \cap E[f]$. This can be done evaluating τ on a basis of $E[f]$ by some discrete logarithm computations (if f is smooth this can be accomplished classically, in the general case one can invoke Shor's polynomial-time quantum algorithm). If $\ker(\tau - d) \cap E[f]$ is cyclic, then $\ker(\tau - d) \cap E[f] = \hat{\phi}$. If not, then one can still recover $\ker \hat{\phi}$ in many cases using [47, Section 4.3.] (there is a small fixable error in this section pointed out in [50]).

The natural idea to counter this attack is to take f to be a non-smooth number, in SCALLOP [36] it is taken to be a large prime. The attack fails as the f-torsion of E is defined over a large extension of $\mathbb{F}_{p^2}$ and one cannot evaluate degree f isogenies without knowing the endomorphism ring of E. On the other hand, when taking a prime conductor order it is not obvious how one can represent the orientation. In SCALLOP this is ensured by writing the natural generator σ as a linear combination of 1 and θ, where θ is an endomorphism of smooth degree. Choosing the orientation first and θ afterwards is in general a challenging task in practice. The key idea in SCALLOP is to somehow choose θ first and the corresponding f afterwards. One possible choice is to take the first few primes of the form $4m + 1$ and represent them as norms of primes $a_k \pm b_k i$ in $\mathbb{Z}[i]$. Then one can take a particular choice for each prime (either plus or minus) and take their product. If the coefficient of i of this product is prime, then it is an appropriate choice for f.

This motivates a hard problem underlying SCALLOP:

Problem 2.1. Let $\phi : E_0 \to E$ a degree f isogeny. Suppose we can evaluate $\sigma = \phi \circ [i] \circ \hat{\phi}$ at any point on E (the cost of the evaluation is the size of the representation of the point). Compute the endomorphism ring of E.

Actually the recent break of pSIDH [21] implies that it is enough to be able to evaluate ϕ at any point, instead of only σ, as then the endomorphism ring of the codomain can be computed in quantum polynomial time.

These design choices already take care of two requirements but in general computing the relation lattice of the class group can be still time consuming. The way this is handled in SCALLOP is to ensure that $f-1$ or $f+1$ is smooth and then the relation lattice can be computed by solving low-order discrete logarithms using the Pohlig-Hellman algorithm.

A different route is taken in SCALLOP-HD [22]. There the authors represent orientations using higher dimensional isogenies. In that setting f can be chosen before choosing θ and then a natural choice is to take $f - 1$ to be a product of large powers of 2 and 3.

Finally we emphasize that in all cases the group action evaluation also entails transporting the orientations (this is not needed in CSIDH as Frobenius provides a canonical orientation by $\mathbb{Z}[\sqrt{-p}]$). In SCALLOP this means that one has to translate the smooth degree endomorphism θ. When the isogeny degree (corresponding to a small norm ideal) and $\deg(\theta)$ are coprime this is just pushing through the kernel of θ through the isogeny. In SCALLOP, parameter choices require the translation through non-coprime degree isogenies. This is more complicated and time consuming; we refer the reader to [36, Section 5.2.] for details.

2.3 Class group computation

Once an $\mathfrak{O}$-orientation of a curve E is known, the cost of the calculation of the action of an ideal $\mathfrak{a}$ of large norm on the isomorphism class of E can be greatly improved by finding a smooth product of primes with small norm $\mathfrak{p}_1^{x_1} \dots \mathfrak{p}_k^{x_k} = (\alpha)\mathfrak{a}$ for some $\alpha \in K$. This means that the class of $\mathfrak{a}$ is the same as that of $\mathfrak{p}_1^{x_1} \dots \mathfrak{p}_k^{x_k}$ in the ideal class group $\mathrm{Cl}(\mathfrak{O})$. Then the action of $\mathfrak{a}$ is simply the composition of the actions of the $\mathfrak{p}_i$, which are significantly easier to compute.

Under the Generalized Riemann Hypothesis (GRH), the class group of an order $\mathfrak{O}$ in a number field is generated by the classes of prime ideals of norm less than $48 \log^2(|\Delta_{\mathfrak{O}}|)$ where $\Delta_{\mathfrak{O}}$ is the discriminant of $\mathfrak{O}$ (a direct consequence of [5, Th. 4]). In practice [13], it was observed that significantly fewer primes were necessary to generate $\mathrm{Cl}(\mathfrak{O})$. Once generators $\mathfrak{p}_1, \dots, \mathfrak{p}_k$ of $\mathrm{Cl}(\mathfrak{O})$ are chosen, our goal to minimize the cost of the evaluation of the action of $\mathfrak{a}$ is to find the shortest exponents $x_1, \dots, x_k$ such that the class $[\mathfrak{a}]$ of $\mathfrak{a}$ in $\mathrm{Cl}(\mathfrak{O})$ is equal to $\prod_i [\mathfrak{p}_i]^{x_i}$. To do that we notice that the exponent vectors $(x_1, \dots, x_k)$ such that $\prod_i [\mathfrak{p}_i]^{x_i} = [1]$ form a Euclidean lattice $\mathcal{L}$ dubbed the *lattice of relations*. Given an initial decomposition of $[\mathfrak{a}]$ with exponent vector $\boldsymbol{x} = (x_1, \dots, x_k)$, we can find a new one by finding a vector $\boldsymbol{u} \in \mathcal{L}$ close to $\boldsymbol{x}$. Then $\boldsymbol{x} - \boldsymbol{u}$ is a new exponent vector of a decomposition of $[\mathfrak{a}]$. If $\boldsymbol{u}$ is the closest vector to $\boldsymbol{x}$, then it is the shortest decomposition possible.

The typical strategy for decomposing $[\mathfrak{a}]$ with respect to a small set of prime generators $(\mathfrak{p}_i)_{i \leq k}$ of $\mathrm{Cl}(\mathfrak{O})$ is to multiply $\mathfrak{a}$ by random short products of the $\mathfrak{p}_i$ and use an ideal reduction technique to obtain $\mathfrak{a}'$ of norm in $O(\sqrt{|\Delta_{\mathfrak{O}}|})$ such that $[\mathfrak{a}'] = [\mathfrak{a}] \cdot \prod_i [\mathfrak{p}_i]^{x_i}$ until $\mathfrak{a}'$ is a product of the $(\mathfrak{p}_i)_{i \leq k}$ (see for example [11, Alg. 2,3]). A similar strategy can be used to compute a generating set of the lattice of relations $\mathcal{L}$: we look for sufficiently many different random decompositions of $\mathfrak{a} = (1)$. When $\mathfrak{O}$ is the maximal order of K (or is non-maximal with a small conductor), the above strategy is the best known technique. For example, this is the case with the signature scheme CSI-FiSh [18] which requires the fast decomposition of random elements in $\mathrm{Cl}(\mathfrak{O})$ to avoid having to use an expensive *rejection sampling* method to ensure security. The best known technique for computing the lattice of relations between a generating set of primes of such an order $\mathfrak{O}$ relies on the class group computation algorithm of Hafner-McCurley [38]. Under the GRH, its complexity is in $L_{|\Delta|}(1/2)$ where

$$L_x(\alpha) = \exp(O((\log x)^{\alpha} (\log \log x)^{1-\alpha})).$$

For objects of size $\log x$, a complexity in $L_x(0)$ means polynomial time, and a complexity in $L_x(1)$ means exponential time. The subexponential nature of the complexity of the Hafner–McCurley algorithm means that for large values of $|\Delta|$, the search for the relation lattice (and decompositions in $\mathrm{Cl}(\mathfrak{O})$) become quickly impractical. Practically speaking, the record computation performed to instantiate CSI-FiSh reached Δ with 512 bits [18].

When the conductor f of the non-maximal order $\mathfrak{O}$ is large, one can significantly reduce the cost of the computation of the lattice of relations and of ideal decomposition in $\mathrm{Cl}(\mathfrak{O})$ by using an algorithm due to Klüners and Pauli [42]. From a high level standpoint, this approach takes advantage of the exact sequence

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

where $\mathfrak{O}_{\mathfrak{p}}$ denotes the localization of $\mathfrak{O}$ at $\mathfrak{p}$. In a nutshell, this means that ideal decomposition (and the search for relations) in $\mathrm{Cl}(\mathfrak{O})$ reduces to ideal decomposition in $\mathrm{Cl}(\mathfrak{O}_K)$ and to the resolution of the Discrete Logarithm Problem (DLP) in the multiplicative groups of the residue fields $\mathfrak{O}_K/\mathfrak{p}$ for $\mathfrak{p} \mid f$ (assuming the factorization of the conductor f is known). See [13, Alg. 2,3] for more details. Note that for a split prime $\mathfrak{p}$, the corresponding instance of the DLP is in a prime field of size $p = \mathcal{N}(\mathfrak{p})$, while in an inert prime, the size of the field is p^2. In the setting of SCALLOP [36], we have $\mathfrak{O}_K = \mathbb{Z}[i]$, which makes all computations in $\mathrm{Cl}(\mathfrak{O}_K)$ easy. On the other hand, no practical implementation beyond 1024-bit discriminants was achieved due to the hardness of the discrete logarithms. The best known algorithms for solving instances of the DLP are variants of the number field sieve (NFS), which has complexity $L_q(1/3)$, where q is the cardinality of the residue field [41]. Practically speaking, we will only use prime fields, where record computations reach q with approximately 800 bits [16].

In summary, the computation of the class group of the order of discriminant $\Delta = -df^2$ where $-d$ is a fundamental discriminant and the factorisation of f is known can be achieved in time

$$L_d(1/2) + \sum_{p|f} L_p(1/3).$$

3 New parameter and design choices

In this section, we propose new instantiations of SCALLOP focusing on both security and efficiency.

The key idea is twofold. Firstly, we use a maximal order with larger class number. Secondly, we choose a conductor f that is not smooth but is also not prime. These methods target concrete efficiency of protocols with security levels equivalent to CSIDH-1024, CSIDH-2048 and CSIDH-4096. Computing the class groups of this size in the CSIDH setting is far out of reach with current classical algorithms and infrastructures. SCALLOP was instantiated for the CSIDH-1024 equivalent case [36], but for the higher security levels, finding a conductor f such that $f \pm 1$ is smooth enough might be more challenging. Furthermore, our goal is to provide more efficient group action evaluations.

Instead of the setting of the Gaussian integers, we start with a quadratic order $\mathbb{Z}[\sqrt{-d}]$ where d is a 256-bit number that will be determined by suitable parameter choices. Class groups of this size can still be computed efficiently in practice [12]. We would like to choose f in a fashion that the discriminant of $\mathbb{Z}[f\sqrt{-d}]$ has 1024/2048/4096 bits. This implies that f should have 384/896/1920 bits.

The high-level idea is as follows. We do not fix d right away, but rather restrict our search to maximal orders that contain an endomorphism with particularly smooth degree of the form $2N^2$. This is ensured by introducing a variable parameter a and looking for pairs a, d such that $a^2 + d = 2N^2$. This quantity is then the norm of the element $a+\sqrt{-d} \in \mathbb{Q}(\sqrt{-d})$. Next, we look for small powers of this element such that the coefficient f of $\sqrt{-d}$ in this power has a particular factorization. We focus on the parameters sets corresponding to CSIDH-1024 and CSIDH-2048.

3.1 Effective orientation from a generator of a suborder

Our generation procedure produces parameters f, d such that we know an element $\omega \in \mathbb{Z}[f\sqrt{-d}]$ of smooth norm, which will correspond to the effective orientation. However, the element ω will in fact never be a generator of $\mathbb{Z}[f\sqrt{-d}]$. Instead it will generate a sub order $\mathbb{Z}[\omega] \subset \mathbb{Z}[f\sqrt{-d}]$, with relative index $g = [\mathbb{Z}[f\sqrt{-d}] : \mathbb{Z}[\omega]]$. Therefore, being able to evaluate ω will not satisfy the original definition of an effective representation [36]. However, Proposition 3.1, shows that this causes no extra problems, as long as we can avoid ideals above primes dividing g.

Proposition 3.1. *Let $\mathfrak{O}$ be an imaginary quadratic order, and let $\mathfrak{O}' \subset \mathfrak{O}$ be a suborder of relative index $g^2 = [\mathfrak{O} : \mathfrak{O}']$. Then given an oriented curve $(E, \iota_E) \in S_{\mathfrak{O}}(p)$, together with an endomorphism ω of E generating $\iota_E(\mathfrak{O}') \subset \mathrm{End}(E)$, one can efficiently evaluate the action of any $\mathfrak{O}$-ideal $\mathfrak{l}$ above $\ell \in O(\log(p))$ on (E, ι_E), provided* $\gcd(\ell, g) = 1$.

Proof. Let $\mathfrak{O} = \mathbb{Z}[\delta]$, and let $\mathfrak{l}$ be an $\mathfrak{O}$ ideal of norm ℓ. Recall that finding the isogeny corresponding to $\mathfrak{l} = (a + \delta, \ell)$ is done by computing $E[[a] + \iota_E(\delta)] \cap E[\ell] = (\widehat{[a] + \iota_E(\delta)})(E[\ell])$. To compute this when only knowing the isogeny corresponding to ω, we use that $g\delta \in \mathfrak{O}'$, hence $g\delta = c + \omega$ for some $c \in \mathbb{Z}$. Then, since $\gcd(\ell, g) = 1$, we have that $\mathfrak{l} = (a + \delta, \ell) = (g(a + \delta), \ell) = (ga + g\delta, \ell) = (ga + c + \omega, \ell)$, and the isogeny corresponding to ℓ can be found in the same way as before, given only the evaluation of ω on $E[\ell]$.

For our application in SCALLOP, it will be sufficient to avoid using ideals above primes dividing $g^2 = [\mathbb{Z}[f\sqrt{-d}] : \mathbb{Z}[\omega]]$ in the basis of the lattice of relations, or ignoring the issue entirely, by additionally searching until all small primes dividing g are non-split in $\mathbb{Z}[\sqrt{-d}]$.

3.2 The CSIDH-1024 case

In this setting we aim to obtain f as the product of three primes of 128 bits each and a very small cofactor. Since we wish to achieve 128-bit security, the natural attacks will fail just as they do when f is prime (thus, there is no compelling reason to take f to be prime). The benefit of this approach is that computing the relation lattice reduces to relatively small finite field discrete logarithm problems.

Fix $N = 2^{129}$; we wish to find a and d such that $d + a^2 = 2N^2 = 2^{259}$. A natural idea would be to take a uniformly at random, compute d accordingly, and raise $a + \sqrt{-d}$ to a small power, hoping that the coefficient of $\sqrt{-d}$ is the product of three prime numbers of size roughly 128 bits (and a possibly very small cofactor). Numbers that are the product of three prime numbers of equal size are relatively dense, but detecting them in practice is potentially hard and time consuming. Instead, we take the more formal approach of computing powers of $a+\sqrt{-d}$ symbolically and expressing the coefficient of $\sqrt{-d}$ in terms of a and d. A script for generating these powers can be found in our implementation repository.

For this specific setting the fourth power, $(a+\sqrt{-d})^4$, represents a particularly suitable choice, as the coefficient of $\sqrt{-d}$ in this quantity is

$$4(a^2 - d)a = 4(2a^2 - 2N^2)a = 8(a - N)(a + N)a,$$

which already splits into three factors and the small cofactor 8. So our goal is find d and a subject to the following two restrictions:

- $a, N - a, a + N$ are all small multiples of 128-bit primes;
- The 128-bit prime factors are all split in $\mathbb{Q}(\sqrt{-d})$.

Remark 3.2. The reason for considering $N - a$ instead of $a - N$ is that $N > a$ as a is chosen to as a 128-bit integer and $N = 2^{129}$

The second condition comes from the fact that we need discrete logarithm computations modulo the prime ideals above those primes (see Section 2.3) which would result in 256-bit discrete logarithm computations (as opposed to 128-bit discrete logarithm computation).

The goal is to sample a from a certain residue class to ensure that whenever $a, a + N, a - N$ are (almost) prime (a precise statement is given in Lemma 3.3), then they are also split in $\mathbb{Q}(\sqrt{-d})$.

Lemma 3.3. *Let $N = 2^{2m+1}$ with $m \geq 0$ and $d = 2N^2 - a^2$ with $0 < a < N$. If $a \equiv 19 \pmod{24}$, then whenever a, $(a+N)/3$ and $N-a$ are prime numbers, they split in $\mathbb{Q}(\sqrt{-d})$*

Proof. Recall that a prime q is split in $\mathbb{Q}(\sqrt{-d})$ if and only if $(\frac{-d}{q}) = 1$, where $(\frac{-d}{q})$ denotes the Legendre symbol. Also note that $a \equiv 19 \pmod{24}$ is equivalent to $a \equiv 1 \pmod{3}$ and $a \equiv 3 \pmod{8}$.

Since $a \equiv 3 \pmod 8$, we have

$$\left(\frac{-d}{a}\right) = \left(\frac{a^2 - 2N^2}{a}\right) = \left(\frac{-2}{a}\right) = \left(\frac{-1}{a}\right)\left(\frac{2}{a}\right) = (-1)(-1) = 1.$$

Similarly, $N - a \equiv -a \equiv 1 \pmod 4$ implies

$$\left(\frac{-d}{N-a}\right) = \left(\frac{(a+N)(a-N) - N^2}{N-a}\right) = \left(\frac{-1}{N-a}\right) = 1.$$

Finally, since $N \equiv 2 \pmod 3$ and $a \equiv 1 \pmod 3$, we see that $a + N$ is divisible by 3. Since $a + N \equiv a \equiv 3 \pmod 4$, we see that $(N + a)/3 \equiv 1 \pmod 4$, so

$$\left(\frac{-d}{(a+N)/3}\right) = \left(\frac{(a+N)(a-N) - N^2}{(a+N)/3}\right) = \left(\frac{-1}{(a+N)/3}\right) = 1.$$

Remark 3.4. Analogous reasoning to the proof of Lemma 3.3 shows that if N is an even power of 2 and $a \equiv 11 \pmod{24}$, then a, $(a + N)/3$ and $N - a$ split again in $\mathbb{Q}(\sqrt{-d})$ when they are prime.

Appropriate parameters can now be generated as follows:

- Set $N = 2^{129}$.
- Sample a random 129-bit number $a \equiv 19 \pmod{24}$.
- Check if a, $(a + N)/3$ and $N - a$ are prime numbers.
- If yes, then set $f = 8(a + N)(N - a)a$ and $d = 2N^2 - a^2$.

3.3 The CSIDH-2048 and CSIDH-4096 case

We could utilize the previous method which would require $896/3 \approx 299$-bit discrete logarithm computations (which is in the feasible range, see Section 2.3) for the CSIDH-2048 case and $1920/3 = 640$-bit discrete logarithm computations for the CSIDH-4096 case. However, we will instead use a slight variation of the previous approach to save on discrete logarithm computations. Rather than taking the fourth power of $a + \sqrt{-d}$, we take the the 12^{th} power and look again at the coefficient of $\sqrt{-d}$, which is given by the expression

$$4a(a^2 - d)(a^2 - 3d)(3a^2 - d)(a^4 - 14a^2d + d^2).$$

Again we let $N = 2^{2m+1}$ be an odd power of 2 of appropriate size and search for d of the form $d = 2N^2 - a^2$. Then the factorization of the expression above becomes

$$128a(a+N)(a-N)(2a^2 - 3N^2)(2a^2 - N^2)(2a^2 - 2aN - N^2)(2a^2 + 2aN - N^2).$$

Lemma 3.5. *Let $N = 2^{2m+1}$ with $m \geq 0$ and $d = 2N^2 - a^2$ with $0 < a < N/\sqrt{2}$. Let*

$$P = \frac{N^2}{2} - a^2, \quad Q = \frac{3N^2 - 2a^2}{10}.$$

If $a^2 \equiv 1 \pmod{30}$, then whenever P and Q are prime numbers, they split in $\mathbb{Q}(\sqrt{-d})$.

Proof. Note that $a^2 \equiv 1 \pmod{30}$ if and only if a is odd and a^2 is congruent to 1 modulo both 3 and 5. The first of these properties is equivalent to $a^2 \equiv 1 \pmod{8}$ (so we actually obtain $a^2 \equiv 1 \pmod{120}$).

Clearly P is an integer. Since $3N^2 \equiv 2 \pmod{10}$ and $a^2 \equiv 1 \pmod{10}$, Q is also an integer. Since $a^2 < N^2/2$, we see that both P and Q are positive.

We have $-d = -3N^2/2 - P = -6 \cdot (2^{2m})^2 - P$, so

$$\left(\frac{-d}{P}\right) = \left(\frac{-1}{P}\right)\left(\frac{2}{P}\right)\left(\frac{3}{P}\right).$$

Now a is odd, so $P \equiv -a^2 \equiv -1 \pmod{8}$ (and hence also $P \equiv -1 \pmod{4}$). Furthermore, $N^2/2 \equiv 2 \pmod{3}$ and $a^2 \equiv 1 \pmod{3}$ imply $P \equiv 1 \pmod{3}$. It follows that

$$\left(\frac{-1}{P}\right) = -1, \quad \left(\frac{2}{P}\right) = 1, \quad \left(\frac{3}{P}\right) = -\left(\frac{P}{3}\right) = \left(\frac{1}{3}\right) = -1,$$

so $\left(\frac{-d}{P}\right) = 1$. Similarly, $-d = -N^2/2 - 5Q = -2 \cdot (2^{2m})^2 - 5Q$. We have $5Q \equiv -a^2 \equiv -1 \pmod{8}$, and hence $Q \equiv 3 \pmod{8}$. It follows that

$$\left(\frac{-d}{Q}\right) = \left(\frac{-2}{Q}\right) = \left(\frac{-1}{Q}\right)\left(\frac{2}{Q}\right) = (-1)(-1) = 1.$$

Lemma 3.6. *Let $N = 2^{2m+1}$ with $m \geq 0$ and $d = 2N^2 - a^2$ with $(\sqrt{3}-1)N/2 < a < N$. Let*

$$R = a^2 + aN - \frac{N^2}{2}, \quad S = \frac{N^2/2 + aN - a^2}{3}.$$

If $a \equiv 7 \pmod{12}$, then whenever R and S are prime numbers, they split in $\mathbb{Q}(\sqrt{-d})$.

Proof. The congruence condition on a yields $a \equiv 3 \pmod{4}$ and $a \equiv 1 \pmod{3}$.

Clearly R is an integer, and since $a \equiv 1 \pmod{3}$ and $N \equiv 2 \pmod{3}$, we see that S is also an integer.

We have $R = (a + N/2)^2 - 3N^2/4$ which is positive because of the lower bound on a. Moreover, $S > N^2/6 > 0$ as $a < N$.

Now $-d = R - N(a + 3N/2) = R - 2 \cdot (2^m)^2(a + 3N/2)$, so

$$\left(\frac{-d}{R}\right) = \left(\frac{-1}{R}\right)\left(\frac{2}{R}\right)\left(\frac{a + 3N/2}{R}\right).$$

Since $R \equiv a^2 \equiv 1 \pmod 8$, we have

$$\left(\frac{-1}{R}\right) = \left(\frac{2}{R}\right) = 1, \quad \left(\frac{a+3N/2}{R}\right) = \left(\frac{R}{a+3N/2}\right).$$

It is easy to verify that $R = (a+3N/2)(a-N/2) + N^2/4$, so $\left(\frac{R}{a+3N/2}\right) = 1$. Hence $\left(\frac{-d}{R}\right) = 1$.

Similarly, $-d = -N(3N/2-a) - 3S = -2 \cdot (2^m)^2(3N/2-a) - 3S$, where we note that $3N/2 - a > 0$ as $a < N$. So

$$\left(\frac{-d}{S}\right) = \left(\frac{-1}{S}\right)\left(\frac{2}{S}\right)\left(\frac{3N/2-a}{S}\right).$$

Since $3S \equiv -a^2 \equiv -1 \pmod 8$, we have $S \equiv -3 \pmod 8$, so $S \equiv 1 \pmod 4$ and we obtain

$$\left(\frac{-1}{S}\right) = 1, \quad \left(\frac{2}{S}\right) = -1, \quad \left(\frac{3N/2-a}{S}\right) = \left(\frac{S}{3N/2-a}\right),$$

and hence $\left(\frac{-d}{S}\right) = -\left(\frac{S}{3N/2-a}\right)$.

Again one readily checks that $3S = (3N/2-a)(a+N/2) - N^2/4$. Since $3N/2 - a \equiv -a \equiv 1 \pmod 4$, we have

$$\left(\frac{3S}{3N/2-a}\right) = \left(\frac{-1}{3N/2-a}\right) = 1$$

and

$$\left(\frac{3}{3N/2-a}\right) = \left(\frac{3N/2-a}{3}\right) = \left(\frac{-a}{3}\right) = \left(\frac{-1}{3}\right) = -1.$$

Overall, we get

$$\left(\frac{-d}{S}\right) = -\left(\frac{S}{3N/2-a}\right) = -\left(\frac{3S}{3N/2-a}\right)\left(\frac{3}{3N/2-a}\right) = -1(-1) = 1.$$

The numerical values of the constants in the bounds on a relative to N appearing in Lemmas 3.5 and 3.6 are $1/\sqrt{2} \approx 0.707$ and $(\sqrt{3}-1)/2 \approx 0.336$.

Combining the congruence conditions on a in Lemmas 3.5 and 3.6 yields $a \equiv 19$ or $31 \pmod{60}$. In conjunction with the restriction $a \equiv 19 \pmod{24}$ from Lemma 3.5, we would require $a \equiv 19$ or $91 \pmod{120}$. However, even if any of the smaller factors a, $N-a$ and $(N+1)/3$ are not prime and contain non-split prime factors, the corresponding discrete log computations are negligible compared to the cost of the discrete log extraction modulo P, Q, R and S.

As in the CSIDH-1024 case, we can now generate suitable parameters for CSIDH-2048 and CSIDH-4096 as follows (note that $30720 = 2^{11} \cdot 3 \cdot 5$).

– Set $N = 2^{129}$ for CSIDH-2048 or $N = 2^{175}$ for CSIDH-4096.
– Sample a random number $a \equiv 19$ or $31 \pmod{60}$ with $(\sqrt{3}-1)N/2 < a < N/\sqrt{2}$.
– Check if P, Q, R, S as given in Lemmas 3.5 and 3.6 are prime numbers.
– If yes, then set $f = 30720a(N+a)(N-a)PQRS$ and $d = 2N^2 - a^2$.

Remark 3.7. It might not be immediately obvious why we set $N = 2^{129}$ for CSIDH-2048 or $N = 2^{175}$ for CSIDH-4096. For CSIDH-2048 we make use of Section 3.1, i.e., when the smooth-norm endomorphism only generates a suborder. In this case we ignore the factor $(a^4 - 14a^2d + d^2)$ for efficiency purposes. Then the size of N is determined by the fact that we need the factor f to have 896 bits. For the CSIDH-4096 case we would like to optimize the precomputation time, hence we would like less costly discrete logarithm computations and we utilize an endomorphism that generates the entire order. The size of N is then determined by the fact that we need the factor f to have 1920 bits.

Remark 3.8. We have $R = (a + N/2)^2 - 3N^3/4$, $3S = (a - N/2)^2 - 3N^2/4$. So for any prime q such that 3 is a quadratic residue modulo q, say $3 \equiv u^2 \pmod{q}$, we have

$$R \equiv \left(a + (1+u)\frac{N}{2}\right)\left(a + (1-u)\frac{N}{2}\right) \pmod{q},$$

with an analogous factorization for S. Thus, if $a \equiv \pm(1 \pm u)N/2 \pmod{q}$, for all four possible sign combinations, then R or S is a multiple of q. For example, if $a \equiv \pm 1$ or $\pm 7 \pmod{11}$, then one of R, S is divisible by 11. This rules out four residue classes modulo q for a for every prime $q \equiv \pm 1 \pmod{12}$. A test for eliminating these congruence class requires the computation of a square root of $3 \pmod{q}$. For small primes q such as 11 and 13, this idea might aid in speeding up the search for suitable parameters, but for large q, such a square root computation is too costly to be useful.

3.4 Security

The security of these new SCALLOP parameters are analyzable similar to the earlier parameter sets, with a few differences taken into consideration for the changes in f and d.

Note that with every parameter set we target 128-bit classical security (and comparable quantum security) as the debate on CSIDH security is about which parameter set achieves 128-bits of security [46],[7].

Generic attacks Recall that a free and transitive group action $\star$ on a group G and set X creates a *hard homogeneous space* if it can be evaluated efficiently and the following two problems are intractable.

Problem 3.9 (Vectorization). Given $x, y \in X$, find $g \in G$ such that $g \star x = y$.

Problem 3.10 (Parallelization). Given $x, g \star x, h \star y \in X$ (for undisclosed $g, h \in G$), find $(g \cdot h) \star x$.

It is a *very hard homogeneous space* if the following problem is also intractable.

Problem 3.11 (Decisional Parallelisation). Given $x, y, u, v \in X$, decide whether there exists some $g \in G$ satisfying $g \star x = y$ and $g \star u = v$.

When the group is $\mathrm{Cl}(\mathfrak{O})$ and the set is $S_{\mathfrak{O}}(p)$ we refer to these as, e.g., the $\mathfrak{O}$-Vectorization problem. It has been shown [54, Theorem 3] that $\mathfrak{O}$-Vectorization reduces in quantum polynomial time to $\mathfrak{O}$-Parallelization.

The fastest classical algorithm for solving the $\mathfrak{O}$-Vectorization problem known [54, Proposition 3] runs in time

$$\log{(p+d)}^{O(1)} \min\left(p^{1/2}, f^{1/2}\right),$$

where $d = \mid \mathrm{disc}\,(\mathfrak{O}) \mid$.

There is a faster quantum attack [54, Proposition 4] on $\mathfrak{O}$-Vectorization, utilizing Kuperberg's algorithm for the Abelian hidden shift problem [43], which asymptotically runs in

$$\log(p)^{O(1)} L_{|\mathrm{disc}(\mathfrak{O})|}(1/2).$$

There are faster quantum algorithms for this problem [20,23,39] which rely on specific group structures, however the class groups from Section 3 have drastically different structures. The general principle in these results is that if the exponent of the group is small, then special purpose algorithms are faster than Kuperberg's algorithm. All SCALLOP variants use class groups with large exponent but our new parameter choice has the extra advantage that the group order is not smooth and has extra flexibility (e.g., one can ensure that the group order has a large prime factor which is useful for threshold schemes [29]).

For the security of the $\mathfrak{O}$-Decisional Parallelization problem, in addition to the above attacks on Vectorization, there also exists distinguishers built from quadratic characters [17,19]. These attacks apply to our case as the order of the class group is even by design. However, one can counter this attack in the usual sense by restricting the group action to the odd part of the class group (this is only necessary for applications where the decisional problem has to be hard).

These characters exist for each divisor $m \mid \mathrm{disc}\,(\mathfrak{O})$, however their evaluation (at least classically) take time polynomial in m. Hence this class of attacks are inefficient when applied to our parameters.

pSIDH type attacks The original SCALLOP construction crucially relies on the hardness of Problem 2.1. The hardness of this problem stems from the following observation.

One is given the evaluation of $\phi \circ \iota \circ \hat{\phi}$ on a point P and one has to find the subgroup generated by $\phi(P)$. If this can be performed for arbitrary P, then Problem 2.1 can be efficiently solved [54, Proposition 7].

Let the order of P be n_P. Then purely working modulo n_P will not be enough information. Indeed, one can precompose ϕ with an endomorphism that commutes with ι and whose degree is congruent to 1 modulo n_P, then the evaluation of the composition does not change whereas the ϕ can change. That is, for integers a, b satisfying $a^2 + b^2 \equiv 1 \mod n_P$, one obtains

$$\left(\phi \circ (a + b\iota) \circ \iota \circ (a - b\iota) \circ \hat{\phi}\right)(P) = \left((a^2 + b^2)\phi \circ \iota \circ \hat{\phi}\right)(P) = \left(\phi \circ \iota \circ \hat{\phi}\right)(P).$$

There are however certain exceptions.

The following counter-example to the above is based on [15, §10]. As in Problem 2.1, consider the isogeny $\phi : E_0 \to E$ of degree f, and endomorphism $\theta \in \mathrm{End}(E_0)$ of degree n_θ, and suppose we can evaluate some $\sigma = \phi \circ \theta \circ \hat{\phi}$ at any point on E. Note this is more general than Problem 2.1, by allowing arbitrary $\theta \in \mathrm{End}(E_0)$. Assume that n_P is prime and coprime to f and n_θ. We show that if the subgroup generated by P is fixed by θ, then this subgroup can be efficiently computed. That is, given $P \in E_0[n_P]$, we compute $[\lambda]\phi(P)$ for some $\lambda \in \mathbb{Z}/n_P\mathbb{Z}^*$.

Suppose $\theta(P) = [a]P$, that is θ fixes the subgroup generated by P. Using the oracle for σ on $E[n_P]$ we can solve discrete logarithms and use linear algebra to compute a point $U \in E[n_P]$ satisfying $\sigma(U) = [fa]U$. We show that $\phi(P)$ is in the subgroup generated by U, and by comparing orders we note that $U = [\lambda]\phi(P)$ for some invertible λ which is our goal. Let $Q \in E_0[n_P]$ be some point independent of P satisfying $\theta(Q) = [b]Q$ for some invertible b which is distinct from a. By coprimality, and the independence of P and Q, we can write $U = \phi([x]P + [y]Q)$ for some x, y. We proceed by showing that $y = 0$.

$$\begin{aligned}[fa]U = \sigma(U) &= \sigma \circ \phi\left([x]P + [y]Q\right) \\ &= [f]\phi \circ \theta\left([x]P + [y]Q\right) = [fxa]\phi(P) + [fyb]\phi(Q).\end{aligned}$$

Multiplying both sides by $f^{-1}a^{-1} \mod n_P$ gives $U = [x]\phi(P) + [yba^{-1}]\phi(Q)$. As we started with $U = \phi([x]P + [y]Q)$, and $ba^{-1} \neq 1$, we conclude that $y = 0$.

So we see that one does get some information on the evaluation of P. This could potentially be combined with some l-adic approach where you glue together local information to get an evaluation of certain points. Note that the above argument is a local one as precomposing ϕ with an endomorphism changes its degree (that should be the fixed f) globally but not locally. Another potential approach to utilize the attack [21] on the NIKE scheme pSIDH [44] directly on isogenies of the form $\phi \circ \iota \circ \hat{\phi}$ as those can be evaluated at any point. There is a similar group action on the set of these types of endomorphisms that is rather closely related to the corresponding isogenies ϕ. However, it is not obvious how to evaluate this action as the approach from [21] does not translate.

In our case using a maximal order with large class number ensures that a similar approach will definitely fail. The reason is that having an oriented curve oriented by a non-maximal order it is hard to find the corresponding curve oriented by the maximal order. One potential avenue here is to go through every curve that is oriented by the maximal order. However, choosing a 256-bit discriminant ensures that such an attack would need 2^{128} iterations.

Torsion-point attacks As mentioned before, choosing f to be smooth would be insecure essentially due the torsion-point attack framework pioneered by [47]. The cost of the attack depends on f-isogeny evaluations and representing points of order f. In our case both have a very large cost as the f-torsion is defined over an extension of the base field $\mathbb{F}_{p^2}$ of degree larger than 2^{128} and the evaluation of f-isogenies requires at least 2^{64} field operations (utilizing [6]) in this large extension. At present there seems to be no practical advantage to take a prime degree isogeny instead of an isogeny which is the product of a few primes.

4 Explicit instantiation of PEARL-SCALLOP

In this section we discuss the implementation details, before providing the timing results, comparing this new parameter set with SCALLOP [36]. Our implementation using SageMath [53] and PARI/GP [52] can be found in the repository `https://www.github.com/biasse/SCALLOP-params`.

4.1 Discriminant generation

We describe our instantiation of CSIDH-1024. Following the method explained in Section 3, we generate a quadratic order $\mathfrak{O}$, providing the class group action, together with an element $\omega \in \mathfrak{O}$, which we will use to evaluate the action. Numerically, reusing the notation from Section 3, we set $N = 2^{129}$, and find that the prime

$$a = 340282366920938463463374607431770911081$$

generates the following values of d and f:

$$\begin{aligned} d &= 18466951 \times 19397359 \times 114814706502110352989273153 \\ &\quad \times 1970795715856882880246375322962354155 1, \text{ and} \\ f &= 2^3 \times 3 \times 340282366920938463463374607431760112581 \\ &\quad \times 340282366920938463463374607431770911081 \\ &\quad \times 340282366920938463463374607431776310331. \end{aligned}$$

Finding this value of a took seconds on a laptop; similarly, a suitable value for a for CSIDH-2048 was found within minutes.

We then select a value of n and let $\ell_1, \ldots, \ell_n$ be the first n split primes that do not dividing the relative conductor $[\mathbb{Z}[\omega] : \mathfrak{O}]$. Subsequently, we choose a prime of the form

$$p = c2^e \prod_{i=1}^{n} \ell_i$$

where e satisfies $\mathrm{nrd}(\omega) = 2^{2e}$, the ℓ_i correspond to the primes in the basis of the lattice of relations, and c is a small cofactor such that p is a prime satisfying $\left(\frac{-\mathrm{disc}(\mathfrak{O})}{p}\right) = 1$. Continuing our example parameters for CSIDH-1024, we have that $n = 75$, $e = 518$ and $c = 817$ generate suitable parameters, for the values a, N used above.

4.2 Computing the relation lattice

Our implementation of the computation of the relation lattice is a variant of the algorithms mentioned in Section 2.3. Let K be an imaginary quadratic field with maximal order $\mathfrak{O}_K$, let $f \geq 1$ be an integer, and let $\mathfrak{O} = \mathbb{Z} + f\mathfrak{O}_K$ be the order of conductor f. Let S be a finite set of prime ideals of K not dividing f. Let $S_{\mathfrak{O}} = \{\mathfrak{p} \cap \mathfrak{O} : \mathfrak{p} \in S\}$, so that every ideal in $S_{\mathfrak{O}}$ is invertible. Recall that the group $\mathfrak{O}_S^\times$ of S-units of $\mathfrak{O}$ is the set of $u \in K^\times$ such that the ideal $u\mathfrak{O}$ is a product of the elements of $S_{\mathfrak{O}}$. Define the morphism $V_S \colon K^\times \to \mathbb{Z}^S$ by

$$V_S(x) = (v_{\mathfrak{p}}(x))_{\mathfrak{p} \in S},$$

where $v_{\mathfrak{p}}(x)$ denotes the $\mathfrak{p}$-adic valuation of x. We write $V_S^{\mathfrak{O}}$ for the restriction of V_S to $\mathfrak{O}_S^\times$. Then the kernel of $V_S^{\mathfrak{O}}$ is the group of units $\mathfrak{O}^\times$ (which is finite and cyclic, and often reduced to $\{\pm 1\}$), and the cokernel of $V_S^{\mathfrak{O}}$ is canonically isomorphic to the subgroup of the class group of $\mathfrak{O}$ generated by the classes of elements in $S_{\mathfrak{O}}$. The image of $V_S^{\mathfrak{O}}$ is the relation lattice of $\mathfrak{O}$ (relative to S).

Remark 4.1. From a relation $v \in V_S(\mathfrak{O}_S^\times)$, we can easily recover a preimage: construct the corresponding ideal and apply lattice reduction with respect to the norm, then the shortest vector will be a generator of the ideal and a preimage of v. Thus it is equivalent to compute the S-unit group and the relation lattice.

The paper [42] focuses on computing the class group of an order, while we are more interested in the relation lattice, so we will use a variant of their algorithm, which has the additional advantage of not requiring any computation of discrete logarithms in the class group once a relation lattice is known for the maximal order. We will use the natural reduction mod f map $\mathrm{Red}_{S,f} \colon \mathfrak{O}_{K,S}^\times \to (\mathfrak{O}_K / f\mathfrak{O}_K)^\times$, which is well-defined since f is not divisible by any ideals of S. We will compute $\mathfrak{O}_S^\times$ using the following well-known fact:

$$\mathfrak{O}_S^\times = \mathrm{Red}_{S,f}^{-1}((\mathbb{Z}/f\mathbb{Z})^\times). \tag{1}$$

The Klüeners–Pauli algorithm is implemented in Magma but it appears to be unable to deal with instances with a maximal order of non-trivial discriminant.

With this setup in place, we can describe our implementation of the computation of the relation lattice relative to a set S.

1. Pick a set S_0 containing S and that provably generates the class group of K, and compute the relation lattice using PARI/GP [52]. There are two algorithms implemented for this task: `bnfinit`, which is designed for number fields of arbitrary degree, and `quadclassunit`, which is a faster implementation for quadratic fields. However, none of these implementations uses sieving, and even for 256 bits discriminants they struggle. We therefore adaptated Pari's implementation (originally implemented by T. Papanikolaou and X. Roblot) of the quadratic sieve factoring algorithm (MPQS), so that it would compute the lattice of relations.
2. Compute the relation lattice of $\mathfrak{O}_K$ relative to S from the relation lattice relative to S_0 by computing an integer kernel.
3. Compute discrete logarithms of a basis of S-units of $\mathfrak{O}_K$ modulo all prime power divisors of f, yielding a description of the map $\mathrm{Red}_{S,f}$ as a matrix. This was done using the Pari implementation of discrete logarithm computations, but we are adapting our implementation to use CADO-NFS [51] since for field size greater than 160 bits the Pari implementation starts to struggle.
4. Compute the relation lattice of $\mathfrak{O}$ relative to S using 1 by computing a kernel modulo the exponent of $(\mathfrak{O}_K/f\mathfrak{O}_K)^\times$.
5. Check that the cardinality of the cokernel of $V_S^{\mathfrak{O}}$ equals the class number of $\mathfrak{O}$, thus proving that S generates the class group of $\mathfrak{O}$.

Let $K = \mathbb{Q}(\sqrt{-d})$ and let S be the set of prime ideals above $\ell_1, \ldots, \ell_n$. The running times of the various steps were as follows, using a single core of an Intel Xeon CPU E5-2623 v3 @ 3.00GHz: Step 1: 45 h; Step 2: 38 s; Step 3: 31 min; Step 4: 12 ms; Step 5: 64 ms. The structure of the class group of the maximal order is

$$\mathrm{Cl}(\mathfrak{O}_K) \cong C_{85291128024656765643024956902338426256} \times C_2^2.$$

The bottleneck for CSIDH-2048 and CSIDH-4096 will be the computation of the discrete logarithms. Our estimation is that the discrete logarithms for CSIDH-2048 would take about 2 months with the Pari implementation; this is sufficient to achieve parameter generation but we prefer to switch to CADO-NFS as our estimate is that it will take a few hours in this case, and CSIDH-4096 would be out of reach with Pari's implementation.

4.3 Lattice reduction

We perform lattice reduction of the relation lattice of $\mathfrak{O}$ relative to S on a 64-core AMD Threadripper 3990X 2.9GHz CPU with 256GB of DDR4 RAM. We used the implementation of the BKZ algorithm [49] from the G6K python library [3] originally presented in [2].

For the lattice reduction corresponding to the CSIDH-1024 parameters, the lattice dimension was $|S| = 75$. We were able to obtain an HKZ reduction of the lattice in 739 CPU seconds. Then the reduction of an input product $\prod_i \mathfrak{p}^{x_i}$ is obtained by using Babai's nearest plane algorithm to find a lattice point u close to $\boldsymbol{x}$, followed by a random walk approach used in CSI-fish [10] due to Doulgerakis, Laarhoven and de Weger [32]. This step takes 3.9 CPU seconds.

Additionally, we performed weighted reductions of the input lattice to account for the cost of evaluating the action of a prime ideal $\mathfrak{p} \in S$. Indeed, since that cost is proportional to $\mathcal{N}(\mathfrak{p})$, the cost of the evaluation of the product $\prod_i \mathfrak{p}_i^{x_i}$ is proportional to $\sum_i x_i \mathcal{N}(\mathfrak{p}_i)$. Therefore, we reduced a weighted lattice where the i-th coordinate is multiplied by $1 + c\mathcal{N}(\mathfrak{p}_i)$, for some constant $0 < c < 1$. This strategy successfully produced short decompositions where the coefficients corresponding to larger primes were significantly smaller than those corresponding to the smaller primes of S. The optimal value of c with respect to the group action evaluation is hard to compute, as it depends on many factors, but it can be estimated for specific implementations.

We do not anticipate that lattice reduction will be the bottleneck to instantiate our system with CSIDH-2048 and CSIDH-4096 parameters.

4.4 Generating the Starting Curve

Recall that we want a starting curve that is $\mathfrak{O}$-orientated, and an efficient way to evaluate the orientation. This is achieved by a tuple (E, P_E, Q_E) where P_E, Q_E are points on E generating smooth isogenies ϕ_P, ϕ_Q, such that their composition $\hat{\phi}_Q \circ \phi_P$ is an element of $\mathfrak{O} \subseteq \mathrm{End}(E)$ (see Section 3.1 for a remark on not using a generator).

We now present a new algorithm for efficiently generating a curve with such an "effective orientation" by any order $\mathfrak{O}$, provided you know an element in the order of smooth norm. Our algorithm is more general, and roughly as effective as the original SetUpCurve algorithm from the original SCALLOP paper [36, Algorithm 1], which only works for orders that are suborders of an order orienting a special p-extremal maximal order (such as $\mathbb{Z} + f\mathbb{Z}[i]$).

We fix a special p-extremal maximal order $\mathcal{O}_0$. The idea of the algorithm is to utilize the fact that the quaternion embedding problem is easily solvable in $\mathcal{O}_0$, provided we hit an easy Cornacchia instance [4, Remark 5.14] (see also [34, Proposition 2]). Hence, we can try different smooth values of g until we can compute an embedding of $\mathbb{Z} + g\mathfrak{O}$ in $\mathcal{O}_0$. We use the heuristic algorithm GenericOrderEmbeddingFactorisation for this purpose, see [34, Algorithm 3]. Given such an embedding, it is then easy to compute an ideal corresponding to the ascending isogeny of degree g. The codomain of this isogeny will then be oriented by $\mathfrak{O}$.

Next, to compute the effective orientation given by a smooth element ω, finding the corresponding kernel generators can easily be done by the standard technique of factoring the ideal in two equal parts, and then translating pack and forth to E_0. This technique is the same as what is present in SQIsign [37].

The whole method is summarized in Algorithm 1.

Remark 4.2. Note that GenerateStartingCurve is somewhat dual to the original method from SCALLOP. In the original method, one computes a *descending* isogeny from a curve with special, p-extremal endomorphism ring oriented by a *superorder*, while in GenerateStartingCurve, we compute the *ascending* isogeny from a curve with special, p-extremal endomorphism ring oriented by a *suborder*.

Algorithm 1 GenerateStartingCurve(γ, p, ω, T)

Input: A generator γ of $\mathfrak{O}$, a prime p such that $\left(\frac{-\mathrm{disc}(\mathfrak{O})}{p}\right) = 1$, an element $\omega \in \mathfrak{O}$ with $\mathrm{norm}(\omega) = L_1 L_2$, L_i smooth and $E[L_i]$ defined over a $\mathbb{F}_{p^2}$, and a powersmooth value $T \gg p/\sqrt{\mathrm{disc}(\mathfrak{O})}$, with $\gcd(L, T) = 1$.

Output: An effectively oriented curve (E, P, Q)

```
Let $i, j, k$ be a basis of $B_{p,\infty}$, such that $i^2 = -q$ and $j^2 = -p$.
Let $\mathcal{O}_0$ be a special, $p$-extremal maximal order in $B_{p,\infty}$, and $E_0$ a supersingular elliptic curve
    with $\mathrm{End}(E_0) \cong \mathcal{O}_0$.
for $n \mid T$ such that $T/n > p/\sqrt{\mathrm{disc}(\mathfrak{O})}$ do
    Set $g := T/n$.
    Set $\delta :=$ GenericOrderEmbeddingFactorisation$(\mathcal{O}_0, \mathrm{trd}(g\gamma), \mathrm{nrd}(g\gamma))$.
    if $\delta \neq \perp$ then
        Break loop.
    end if
end for
Set $I := \mathcal{O}_0\langle g, \delta\rangle$.
Compute $\phi_I$ from $I$ using IdealToIsogeny.
Set $E := \phi_I(E_0)$, and compute $\mathcal{O} := \mathcal{O}_R(I)$.
Let $H_1 := \mathcal{O}\langle L_1, \omega\rangle$, and $H_2 := \mathcal{O}\langle L_2, \bar{\omega}\rangle$.
Translate $[I]^* H_i$ to their kernel generators $K_i$ using IdealToKernel.
Set $P := \phi_I(K_1)$ and $Q = \phi_I(K_2)$.
return $(E, P, Q)$
```

Proposition 4.3. GenerateStartingCurve *is correct and runs in probabilistic polynomial time, under heuristics.*

Proof. First we prove that the first part of the algorithm terminates and is correct. Each run run of GenericOrderEmbeddingFactorisation runs in polynomial time under [34, Heuristic 1, Heuristic 2], and again under [34, Heuristic 2], the number of maximal orders oriented by $\mathbb{Z} + g\mathfrak{O}$ is $O(p)$, hence a solution is expected to exist.

Assume now that an optimal embedding given by δ has been found. $I := \mathcal{O}_0\langle \delta, g\rangle$ corresponds to an ascending isogeny of degree f (it is generated by the unique invertible $(\mathbb{Z} + g\mathfrak{O})$-ideal of norm g). Further, since f is powersmooth of size $O(p/\sqrt{\operatorname{disc}(\mathfrak{O})})$, translating I to its corresponding isogeny ϕ_I using $\mathsf{IdealToIsogeny}$ is efficient.

Assume now that we have a curve E with $\operatorname{End}(E) \cong \mathcal{O}$ oriented by $\mathfrak{O}$. We use the smooth norm ideal $\mathcal{O}\langle\omega\rangle$ to find the efficient orientation. This is done by writing $\mathcal{O}\langle\omega\rangle = \bar{H}_2 \cdot H_1$, where H_i can be efficiently translated, by pulling back to an ideal of $\mathcal{O}_0$, using I, since $\gcd(\operatorname{nrd}(I), \operatorname{nrd}(H_i)) = 1$.

4.5 Evaluating an element

When given an element of $\mathrm{Cl}(\mathfrak{O})$, whose action we wish to evaluate, we first find a smooth representative, as explained in Section 4.3. Following, we wish to evaluate the ideal

$$\mathfrak{a} = \prod_{i=1}^{N} \mathfrak{l}^{e_i}$$

This is, as usual, done by repeatedly applying Algorithm 2 an ideals $\mathfrak{a}_0 \mid \mathfrak{a}$ on the form

$$\mathfrak{a} = \prod_{i=1}^{N} \mathfrak{l},$$

until all of $\mathfrak{a}$ has been evaluated. Since the norm of ω is a power of 2, and specifically, coprime to the primes used in the factor basis, this allows Algorithm 2 to be particularily simple, compared to the equivalent algorithms in the original version of SCALLOP [36, Algorithm 2], or SCALLOP-HD [22, Algorithm 3].

As an optimisation, we also present an even simpler, but probabilistic group action evaluation algorithm below. This is based on a standard CSIDH-optimisation to avoiding sampling points of full order. Since Algorithm 2 requires computing full torsion bases, this optimistic sampling is particularly worthwile in SCALLOP.

4.6 Implementations

We implement a Proof of Concept version of PEARL-SCALLOP with our parameters in C++. The parameters used can be found in the repository. Our implementation applies some well known, standard optimisations: We work with montgomery curves

$$E : y^2 = x^3 + Ax^2 + x,$$

and to compute the 2^e-isogeny corresponding to the orientation, we use the formula for 4-isogenes, together with optimal strategies, following the SIKE documentation [40]. For evaluating the group action, we use Algorithm 3.

Algorithm 2 GroupAction$(\mathfrak{a}, E, P, Q)$

```
Input: An $\mathfrak{O}$-ideal $\mathfrak{a}$ of the form $\prod \mathfrak{l}_i$, an elliptic curve oriented by $\mathfrak{O}$, and points $P, Q \in E$
    generating $\phi_P, \phi_Q$ such that $\widehat{\phi_Q} \circ \phi_P$ is an endomorphism corresponding to an element of $\mathfrak{O}$
    of norm $2^e$
Output: An effectively oriented curve $(E_\mathfrak{a}, P_\mathfrak{a}, Q_\mathfrak{a})$
Let $B_1, B_2$ be a basis of $E[L]$, where $L = \prod \ell_i, \quad \ell_i = \mathrm{nrd}(\mathfrak{l}_i)$
Let $\widehat{\omega} := \widehat{\phi_P} \circ \phi_Q$
Compute $B_1' = \widehat{\omega}(B_1), B_2' = \widehat{\omega}(B_2)$.
for $i \in \{1, \dots, N\}$ do
    Compute $K_i := [L/\ell_i]([\lambda_i]B_1 + B_1'$, where $\mathfrak{l}_i = (\lambda_i + \omega, \ell_i)$.
    if $K_i = \infty$ then
        Compute $K_i := [L/\ell_i]([\lambda_i]B_2 + B_2'$.
    end if
end for
Compute $\phi_\mathfrak{a} : E \to E_\mathfrak{a}$ from its kernel $K = \langle K_1, K_2, \dots, K_N \rangle$.
return $E_\mathfrak{a}, \phi_\mathfrak{a}(P), \phi_\mathfrak{a}(Q)$
```

Algorithm 3 GroupActionOptimistic$(\mathfrak{a}, E, P, Q)$

```
Input: An $\mathfrak{O}$-ideal $\mathfrak{a}$ of the form $\prod \mathfrak{l}_i$, an elliptic curve oriented by $\mathfrak{O}$, and points $P, Q \in E$
    generating $\phi_P, \phi_Q$ such that $\widehat{\phi_Q} \circ \phi_P$ is an endomorphism corresponding to an element of $\mathfrak{O}$
    of norm $2^e$
Output: An effectively oriented curve $(E_\mathfrak{a}', P_{\mathfrak{a}'}, Q_{\mathfrak{a}'})$, and $n(\mathfrak{a}')$, where $\mathfrak{a}' \mid \mathfrak{a}$
Compute $K := [\frac{p+1}{L}]K_0$, where $K_0$ is a random point on $E$.
Let $\widehat{\omega} := \widehat{\phi_P} \circ \phi_Q$
Compute $K' := \widehat{\omega}(K)$
Compute $K_{\mathfrak{a}'} := [\lambda]K + K'$, where $\bar{\mathfrak{a}} = (\lambda + \omega, L)$.
Compute $\phi_{\mathfrak{a}'} : E \to E_{\mathfrak{a}'}$ from its kernel $\langle K_{\mathfrak{a}'} \rangle$
return $E_{\mathfrak{a}'}, \phi_{\mathfrak{a}'}(P), \phi_{\mathfrak{a}'}(Q), L'$, where $L' = n(\mathfrak{a}')$.
```

Table 1. Timings from SCALLOP [36], SCALLOP-HD[22] and PEARL-SCALLOP.

Security level	[36, Section 6.2]	[22, Section 5.6]	This work
CSIDH-512	35 sec	1 min, 28 sec	30 sec
CSIDH-1024	12 min, 30 sec	19 min	58 sec
CSIDH-2048	-	-	TBD
CSDIH-3072	-	-	TBD
CSDIH-4096	-	-	TBD

We give timings for evaluating a group element, and compare with the timings reported in SCALLOP and SCALLOP-HD in Table 1.

Note that all three implementations are unoptimised Proof-of-concept implementations, hence the timings only give a rough idea of the picture. In particular, the timings in SCALLOP-HD is based on a SageMath [53] implementation, rather than C++. However, comparing the efficiency of PEARL-SCALLOP and SCALLOP-HD is also easy on a theoretical level: The choices of prime field, the group action evaluation can be made almost identical in both cases, and the group action evaluation is very similar, except in SCALLOP-HD it requires the evaluation of a $(2^e, 2^e)$-isogeny between abelian surfaces, while in PEARL-SCALLOP, it relies on the evaluation of a 2^e-isogeny between elliptic curves.

5 Conclusion

In this work, we have presented PEARL-SCALLOP, a new way of instantiating an efficient cryptographic group action, based on SCALLOP [36]. Compared to SCALLOP, this work is easier to instantiate for higher security levels, is significantly more efficient in terms of group action evaluation, and is based on a different hardness assumption.

Compared to SCALLOP-HD [22], another efficient cryptographic group action based on SCALLOP, this work is again more efficient in terms of group action evaluation, and based on a different hardness assumption, though SCALLOP-HD is easier to instantiate for even higher security levels. However, in this work, we provide practical instantiations for security level equivalent to CSIDH-4096, arguing that PEARL-SCALLOP is currently the ideal choice for efficiency, while also being possible to instantiate for secure parameter levels.

References

1. Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. Cryptographic group actions and applications. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II*, volume 12492 of *Lecture Notes in Computer Science*, pages 411–439. Springer, 2020.
2. Martin R. Albrecht, Léo Ducas, Gottfried Herold, Elena Kirshanova, Eamonn W. Postlethwaite, and Marc Stevens. The general sieve kernel and new records in lattice reduction. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part II*, volume 11477 of *Lecture Notes in Computer Science*, pages 717–746. Springer, 2019.
3. Martin R. Albrecht, Léo Ducas, Gottfried Herold, Elena Kirshanova, Eamonn W. Postlethwaite, and Marc Stevens. The General Sieve Kernel (G6K), 2019.

4. Sarah Arpin, James Clements, Pierrick Dartois, Jonathan Komada Eriksen, Péter Kutas, and Benjamin Wesolowski. Finding orientations of supersingular elliptic curves and quaternion orders. *IACR Cryptol. ePrint Arch.*, page 1268, 2023.
5. Eric Bach. Explicit bounds for primality testing and related problems. *Math. Comp.*, 55(191):355–380, 1990.
6. Daniel J. Bernstein, Luca De Feo, Antonin Leroux, and Benjamin Smith. Faster computation of isogenies of large prime degree. *ANTS*, 2020.
7. Daniel J Bernstein, Tanja Lange, Chloe Martindale, and Lorenz Panny. Quantum circuits for the csidh: optimizing quantum evaluation of isogenies. In *Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part II 38*, pages 409–441. Springer, 2019.
8. Ward Beullens, Samuel Dobson, Shuichi Katsumata, Yi-Fu Lai, and Federico Pintore. Group signatures and more from isogenies and lattices: Generic, simple, and efficient. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 95–126. Springer, 2022.
9. Ward Beullens, Shuichi Katsumata, and Federico Pintore. Calamari and Falafl: logarithmic (linkable) ring signatures from isogenies and lattices. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 464–492. Springer, 2020.
10. Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 227–247. Springer, 2019.
11. J.-F. Biasse and C. Fieker. Subexponential class group and unit group computation in large degree number fields. *LMS Journal of Computation and Mathematics*, 17:385–403, 1 2014.
12. Jean-François. Biasse. Improvements in the computation of ideal class groups of imaginary quadratic number fields. *Adv. in Math. of Comm.*, 4(2):141–154, 2010.
13. Jean-François Biasse, Claus Fieker, and Michael J. Jacobson Jr. Fast heuristic algorithms for computing relations in the class group of a quadratic order, with applications to isogeny evaluation. *LMS Journal of Computation and Mathematics*, 19(A):371â€“390, 2016.
14. Xavier Bonnetain and André Schrottenloher. Quantum security analysis of CSIDH. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 493–522. Springer, 2020.
15. Paul Bottinelli, Victoria de Quehen, Chris Leonardi, Anton Mosunov, Filip Pawlega, and Milap Sheth. The dark SIDH of isogenies. Cryptology ePrint Archive, Paper 2019/1333, 2019.
16. Fabrice Boudot, Pierrick Gaudry, Aurore Guillevic, Nadia Heninger, Emmanuel Thomé, and Paul Zimmermann. Comparing the difficulty of factorization and discrete logarithm: a 240-digit experiment. In *Advances in cryptology – CRYPTO 2020. 40th annual international cryptology conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17–21, 2020. Proceedings. Part II*, pages 62–91. Cham: Springer, 2020.
17. Wouter Castryck, Marc Houben, Frederik Vercauteren, and Benjamin Wesolowski. On the decisional Diffie–Hellman problem for class group actions on oriented elliptic curves. *CoRR*, abs/2210.01160, 2022. To appear in the proceedings of the Fifteenth Algorithmic Number Theory Symposium, ANTS-XV.

18. Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: an efficient post-quantum commutative group action. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 395–427. Springer, 2018.
19. Wouter Castryck, Jana Sotáková, and Frederik Vercauteren. Breaking the decisional Diffie–Hellman problem for class group actions using genus theory. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020*, volume 12171 of *Lecture Notes in Computer Science*, pages 92–120. Springer, 2020.
20. Wouter Castryck and Natan vander Meeren. Two remarks on the vectorization problem. *Cryptology ePrint Archive*, 2022.
21. Mingjie Chen, Muhammad Imran, Gábor Ivanyos, Péter Kutas, Antonin Leroux, and Christophe Petit. Hidden stabilizers, the isogeny to endomorphism ring problem and the cryptanalysis of pSIDH. *arXiv preprint arXiv:2305.19897*, 2023.
22. Mingjie Chen and Antonin Leroux. SCALLOP-HD: group action from 2-dimensional isogenies. *Cryptology ePrint Archive*, 2023.
23. Andrew M. Childs and Wim van Dam. Quantum algorithms for algebraic problems. *Reviews of Modern Physics*, 82(1):1, 2010.
24. Leonardo Colò and David Kohel. Orienting supersingular isogeny graphs. *Number-Theoretic Methods in Cryptology 2019*, 2019.
25. Jean Marc Couveignes. Hard homogeneous spaces. *IACR Cryptology ePrint Archive*, 2006:291, 2006.
26. David A. Cox. *Primes of the form $x^2 + ny^2$ — Fermat, class field theory, and complex multiplication*. AMS Chelsea Publishing, Providence, RI, third edition, [2022] ©2022. With contributions by Roger Lipsett.
27. Pierrick Dartois, Antonin Leroux, Damien Robert, and Benjamin Wesolowski. SQISignHD: New dimensions in cryptography. *Cryptology ePrint Archive*, 2023.
28. Luca De Feo and Steven D. Galbraith. SeaSign: Compact isogeny signatures from class group actions. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 759–789. Springer, 2019.
29. Luca De Feo and Michael Meyer. Threshold schemes from isogeny assumptions. In *IACR International Conference on Public-Key Cryptography*, pages 187–212. Springer, 2020.
30. Victoria de Quehen, Péter Kutas, Chris Leonardi, Chloe Martindale, Lorenz Panny, Christophe Petit, and Katherine E Stange. Improved torsion-point attacks on sidh variants. In *Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part III 41*, pages 432–470. Springer, 2021.
31. Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Trans. Inf. Theory*, 22(6):644–654, 1976.
32. Emmanouil Doulgerakis, Thijs Laarhoven, and Benne de Weger. Finding closest lattice vectors using approximate voronoi cells. In Jintai Ding and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019, Chongqing, China, May 8-10, 2019 Revised Selected Papers*, volume 11505 of *Lecture Notes in Computer Science*, pages 3–22. Springer, 2019.
33. Kirsten Eisenträger, Sean Hallgren, Kristin Lauter, Travis Morrison, and Christophe Petit. Supersingular isogeny graphs and endomorphism rings: Reductions and solutions. In

Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, pages 329–368. Springer International Publishing, 2018.

34. Jonathan Komada Eriksen and Antonin Leroux. Computing orientations from the endomorphism ring of supersingular curves and applications. *IACR Cryptol. ePrint Arch.*, page 146, 2024.
35. Jonathan Komada Eriksen, Lorenz Panny, Jana Sotáková, and Mattia Veroni. Deuring for the people: Supersingular elliptic curves with prescribed endomorphism ring in general characteristic. *Cryptology ePrint Archive*, 2023.
36. Luca De Feo, Tako Boris Fouotsa, Péter Kutas, Antonin Leroux, Simon-Philipp Merz, Lorenz Panny, and Benjamin Wesolowski. SCALLOP: scaling the CSI-FiSh. In *IACR International Conference on Public-Key Cryptography*, pages 345–375. Springer, 2023.
37. Luca De Feo, Antonin Leroux, Patrick Longa, and Benjamin Wesolowski. New algorithms for the deuring correspondence - towards practical and secure SQISign signatures. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*, volume 14008 of *Lecture Notes in Computer Science*, pages 659–690. Springer, 2023.
38. James L. Hafner and Kevin S. McCurley. A rigorous subexponential algorithm for computation of class groups. *J. Am. Math. Soc.*, 2(4):837–850, 1989.
39. Gábor Ivanyos. On solving systems of random linear disequations. *arXiv preprint arXiv:0704.2988*, 2007.
40. David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, Vladimir Soukharev, David Urbanik, Geovandro Pereira, Koray Karabina, and Aaron Hutchinson. SIKE. Technical report, National Institute of Standards and Technology, 2022. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions.
41. Antoine Joux, Andrew Odlyzko, and Cécile Pierrot. The past, evolving present, and future of the discrete logarithm. In *Open problems in mathematics and computational science. Based on the presentations at the conference, Istanbul, Turkey, September 18–20, 2013*, pages 5–36. Cham: Springer, 2014.
42. J. Klüners and S. Pauli. Computing residue class rings and Picard groups of orders. *Journal of Algebra*, 292(1):47 – 64, 2005.
43. Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM Journal on Computing*, 35(1):170–188, 2005.
44. Antonin Leroux. A new isogeny representation and applications to cryptography. *ASIACRYPT*, 2022.
45. Hiroshi Onuki. On oriented supersingular elliptic curves. *Finite Fields and Their Applications*, 69:101777, 2021.
46. Chris Peikert. He gives C-sieves on the CSIDH. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 463–492. Springer, 2020.
47. Christophe Petit. Faster algorithms for isogeny problems using torsion point images. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 330–353. Springer, 2017.
48. Alexander Rostovtsev and Anton Stolbunov. Public-key cryptosystem based on isogenies. Cryptology ePrint Archive, Report 2006/145, 2006.

49. Claus-PSCIeter Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Math. Program.*, 66(2):181–199, September 1994.
50. Francesco Sica. Two remarks on torsion-point attacks in isogeny-based cryptography. *Cryptology ePrint Archive*, 2023.
51. The CADO-NFS Development Team. CADO-NFS, an implementation of the number field sieve algorithm, 2017. Release 2.3.0.
52. The PARI Group, Univ. Bordeaux. *PARI/GP version `2.16.1`*, 2022. available from `http://pari.math.u-bordeaux.fr/`.
53. The Sage Developers. *SageMath, the Sage Mathematics Software System (version 9.7)*, 2022. `https://sagemath.org`.
54. Benjamin Wesolowski. Orientations and the supersingular endomorphism ring problem. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022*, volume 13277 of *Lecture Notes in Computer Science*, pages 345–371. Springer, 2022.