On the efficient representation of isogenies

A survey for NuTMiC 2024

DAMIEN ROBERT

Abstract. We survey different (efficient or not) representations of isogenies, with a particular focus on the recent "higher dimensional" isogeny representation, and algorithms to manipulate them.

1. Introduction

The field of isogeny based cryptography changed drastically following the SIDH attacks [CD23; MMPPW23; Rob23b]. Indeed, the main byproduct of these attacks is a new efficient representation of isogenies, which we will call the *higher dimensional representation* or *HD representation*. This new representation quickly found cryptographic applications: SQIsignHD [DLRW24], FESTA [BMP23] and QFESTA [NO23], the Deuring VRF [Ler23b], an isogeny VDF [DMS23], SCALLOP-HD [CLP24], IS-CUBE [Mor23], LIT-SiGamal [Mor24], SILBE [DFV24], POKE [Bas24], SQIsign2d (West and East) [BDD+24; NO24], SQIPrime [DF24]…It gave rise to new methods to convert ideals to isogenies, both in the supersingular case [Ler23b; ON24; BDD+24] and in the oriented case [PR23b]. Apart from protocols, the HD representation was used to obtain new or better security reductions for isogeny based cryptography [MW23; ACD+23; PW24; ES24], and also better (classical) security reductions for the DLP between isogeneous elliptic curves [Gal24]. Finally, it also gave applications in number theory: computing the endomorphism ring of an ordinary elliptic curve E in polynomial time (if we are provided with the factorisation of the discriminant Δ_E), point counting for $E/\mathbb{F}_{p^n}$ in $\widetilde{O}(n^2 \log^{O(1)} p)$, canonical lift of an ordinary $E/\mathbb{F}_{p^n}$ to precision m in $\widetilde{O}(mn \log^{O(1)} p)$, and new algorithms to compute modular polynomials [Rob22b; KR24].

1.1. History.

The groundbreaking idea to attack the SIDH supersingular elliptic curve cryptosystem [JD11; DJP14] using higher dimensional isogenies is due to Castryck and Decru, and independently to Maino and Martindale.

In 2022, Decru was working on building a VDF (Verifiable Delay Function) using isogenies in dimension 2. He realised that exploiting isogenies $E_1 \times E_2 \to E_1' \times E_2'$ between products of elliptic curves could potentially be used to attack the SIDH cryptosystem. He fully worked this attack out with Castryck, and in their preliminary article [CD22] (which was later published as [CD23]) they gave a working Magma implementation to break SIDH in (heuristic) polynomial time from the special starting curve E_0. In that attack, they made a crucial use of a technical result (now called Kani's lemma) due tu Kani [Kan97, § 2, Proof of Th. 2.3], and which will be key to derive the HD representation. The reason the attack works over the special curve E_0 is that it contains many known endomorphisms: notably endomorphisms of the form $a + bi$ which are of degrees $a^2 + b^2$. In the first version of their paper, Castryck and Decru sketched how their attack could be extended to an arbitrary

Date: November 8, 2024.

starting supersingular curve E, but that it would be unlikely to be practical. De Feo[1] pointed out that this still gave an (heuristic) subexponential attack for an arbitrary starting curve E with unknown endomorphisms. And Wesolowski, in a note, also explained how Castryck and Decru's polynomial attack could extend to such an arbitrary E, provided that $\mathrm{End}(E)$ was known.

Independently, Maino, who had visited the Cosic group in Leuven earlier, during which he had discussed the dimension 2 VDF with Castryck and Decru, also had realised that dimension 2 isogenies could be used to attack SIDH. He first sketched with Martindale an effective attack path in [MM22], and then they gave a more fledged out attack, along with a Sage implementation, and with contributions by Panny, Pope and Wesolowski (and extra help by De Feo and Oudompheng) in [MMPPW23]. The original attack of [CD23], used a decisional version: they guess part of Alice's secret isogeny, then check using higher dimensional isogenies if that guess is correct. This allowed them to reconstruct Alice's isogeny step by step. By contrast the version of Maino and Martindale was direct: they used dimension 2 isogenies to directly recover Alice's secret isogeny. More precisely, they focused on an arbitrary starting curve E, and after a precomputation step to select appropriate parameters, they gave a direct algorithm to recover the secret isogenies[2]. The complexity of that direct key recovery depended on the parameters found, but the same heuristic complexity analysis as done by De Feo for Castryck and Decru's attack showed that it gave a subexponential attack. Combining the direct attack of Maino-Martindale with the exploitation of the special known endomorphisms of E_0 as in Castryck and Decru's version considerably improved the practical key recovery attack when the starting curve was E_0. In the SageMath reimplementation of the Magma code of Castryck and Decru, contributed by Oudompheng, Panny and Pope among others, incorporating this direct key recovery made the attack go from minutes or hours to seconds or minutes[3] (depending on the security parameters). This was the first hint that these attacks could potentially be applied for constructive use.

Once the idea of using dimension 2 isogenies to attack SIDH was introduced, it was natural to look at whether using even higher dimensions could improve these attacks: notably for a starting curve E with unknown endomorphisms (Wesolowski had the same idea[1]). In [Rob23b], we explained how to combine Kani's lemma as used in [CD23; MMPPW23] with Zarhin's trick [Zar74] to attack SIDH in (proven) polynomial time even with a random starting curve E, by going to dimension up to $g = 8$. Indeed, the main obstacle preventing the polynomial time attack of [CD23] on the special curve E_0 to be applied to a random curve E is the lack of known endomorphisms (apart from integer multiplications) on E. Zarhin's trick solves this problem: we can always build many endomorphisms on E^2 and E^4 respecting the product principal polarisation by using suitable integer matrices. So we could apply the Castryck-Decru-Maino-Martindale attack replacing E by E^4, and using dimension 8 isogenies rather than dimension 2 isogenies. By luck, we had already used Zarhin's trick in our algorithm [CR15] to compute higher dimensional isogenies, so it was natural to apply it to the SIDH attack too. This also required to extend Kani's technical lemma from elliptic curves to abelian varieties, but that extension was completely straightforward[4].

[1] Private communication

[2] After the first version of [CD23] was published, this direct key recovery improvement had also been independently found out by Oudompheng, Petit and Wesolowksi.

[3] With the low level C and Rust implementation we now have of the improved formulas for dimension 2 2^e-isogenies [DMPR24], what would take a few seconds in 2022 would now take only a few ms!

[4] It also helped that we were already familiar with some of Kani's work which we had used to understand the denominators of Hilbert modular polynomials [MR19; Rob21, § 5.3.6].

In [Rob23b], the main algorithmic tool used for the attack was the following *embedding lemma*: for any $N > n$ coprime to n, an n-isogeny $\phi : E_1 \to E_2$ can be efficiently embedded into an N-isogeny $\Phi : A \to B$ in higher dimension $g = 2, 4, 8$, provided we know how ϕ acts on the N-torsion of E_1. More generally, the embedding lemma allows to embed an n-isogeny in dimension g into an N-isogeny in dimension $2g, 4g, 8g$. (For the embedding lemma with special curves like E_0, then g can often be smaller than for a generic curve, see Remark 5.17.)

In the conclusion of [Rob23b], we asked the following question: "This tool allows one to break SIDH efficiently in all cases. Can it also be used to build new isogeny based cryptosystems?" One week after the first version of that article, we made in [Rob22a] the rather obvious remark that by taking N powersmooth above, the embedding lemma proves that any isogeny admitted an efficient representation, meaning a representation taking polynomial (in $\log n$ and $\log q$) space and time: this is the HD representation (although it was not given a name in that paper)! The conclusion of that article was: "The method presented above shows that the efficient computation of isogenies for higher dimensional abelian varieties has interesting algorithmic applications to elliptic curves. Hopefully, this is the start of many new results in this direction."

And indeed, as illustrated by all the applications above, the HD representation quickly found lots of use. Still, these new applications are not immediate: the HD representation shows that if we know the evaluation of ϕ on sufficiently many nice points, we can efficiently evaluate ϕ everywhere. But it seems that we have a bootstrapping problem: how can we evaluate ϕ on these nice points to begin with? To answer this question, many new algorithms have been developed to work directly with these HD representations: divisions, duals, splittings, pushforwards…

1.2. **A survey.** Almost two years after the SIDH attack, the NuTMiC (Number-Theoretic Methods in Cryptology) conference held in Szczecin seemed like a nice occasion to do a survey on the use of the HD representation and the algorithms developed to work with them.

Of course, the HD representation is not the only useful representation of an isogeny, so for this survey we will try to briefly explain the many ways we can represent an isogeny, and how we can convert between all these different representations.

In his invited talk for Eurocrypt 2024, Castryck gave a wonderful talk on: "An Attack Became a Tool: Isogeny-based Cryptography 2.0". The aim of this survey is to give an overview of the algorithms that have been developed for this renewal of isogeny based cryptography, so that hopefully they become accessible to a broader audience.

We apologize for the length of this survey, which is longer than initially expected.

1.3. **Thanks.** We thank Andrea Basso for several useful comments, and Julien Soumier for pointing out some typos.

1.4. **Outline.** In Section 2, we give an overview of the different isogeny representations we will describe in this survey, what we mean by an efficient representation, and an overview of the algorithms we now have to work with efficient representations that may not be given (anymore!) by kernel generators of smooth order.

In Section 3, we survey the "standard" isogeny representations, meaning the ones that do not use higher dimension.

In Section 4, we survey the ideal representations of "horizontal" isogenies. Efficient ideal to isogeny algorithms have been key to develop efficient isogeny based cryptosystems (like SQIsign [DKLPW20; DLLW23]). The original version of SQIsign relied on the KLPT algorithm [KLPT14] for this conversion. But newer versions (SQIsignHD, SQIsign2d) have

switched to an ideal to isogeny algorithm relying on the HD representation. We give a very brief overview of these different methods, both for the supersingular case and the oriented case.

In Section 5, we introduce in more details the HD representation. And then in Section 6, we describe algorithms to work with efficient representations of isogenies.

Finally in Section 7 we give a list of open questions.

In the appendices, we treat some technical subjects in more details. In Appendix A we look at the accessible torsion in an elliptic curve (or abelian variety). In Appendix B we explain how to relax the torsion requirement for the HD representation from $N > n$ to $N^2 > n$. In Appendix C, we give more details on the different generations of the ideal to isogeny algorithms in the supersingular case, and how they each gave rise to improvements to SQIsign. In Appendix D we give a geometric interpretation of the usual class group exact sequence for a non maximal quadratic order R, and explain the relationship between this geometric interpretation and the level structure encoded by going up isogenies. Finally in Appendix E we give some technical remarks on the Kodaira-Spencer isomorphism, which we need for the deformation representation and modular representation of isogenies in higher dimension.

2. Overview

In Section 2.1 we define what we mean by an efficient isogeny. Then in Section 2.2 we quickly review the "classical" representations of an isogeny, in Section 2.3 the ideal representations, and then we look at the HD representation in Section 2.4. We survey existing algorithms on the HD representation in Section 2.5.

2.1. Efficient representation of an isogeny.

Let $\phi : E_1 \to E_2$ be an isogeny of degree n between elliptic curves defined over a field k (we will also say that ϕ is an n-isogeny). A representation of ϕ is any data that encodes the domain E_1, the codomain E_2, the degree n, along with a way (an algorithm) to evaluate the image by ϕ of a point $P \in E_1(k')$, where k'/k is a field extension. It can be useful to relax the condition that P is defined over a field, and allow for points over a k-algebra, notably to be able to work with formal points and their images.

In this survey paper, we will contend ourselves to work with this informal definition. For a more formal definition of the representation of an isogeny, we refer to [Ler22a].

For simplicity, we will assume that k is a finite field $k = \mathbb{F}_q$, $q = p^d$, and that the degree n of ϕ is prime to p. We will also stick to isogenies between elliptic curves, although many of the constructions we will introduce generalise to principally polarised abelian varieties (and some of our constructions for elliptic curves will actually use abelian varieties of dimension $g > 1$ as explained in Section 1). In practice, we will also always work with Weierstrass equations for E_1, E_2. For most applications, one can also assume that ϕ is cyclic, that is that its kernel is isomorphic as a group to $\mathbb{Z}/n\mathbb{Z}$, although for some applications it will be convenient to treat all cases.

We will say that an isogeny representation is *compact*, or *space efficient*, if the data to encode it is polynomial in $\log n$ and $\log q$. We will say that it is *efficient* if, from the encoded data, it can evaluate the image of a point $P \in E_1(\mathbb{F}_{q'})$ in time polynomial in $\log n$ (so polylogarithmic in n) and $\log(q')$.

Remark 2.1 (Efficient versus practical representations)**.** With our definitions, an isogeny representation that allows to compute the image of a point in $O(\log^C n)$ arithmetic operations will be an efficient representation, even if C is a big constant. For *practical* applications of

isogeny representations, rather than theoretical results, we will want our representations to be computable in *practice*, and as *fast* as possible.

For instance, the HD representation embeds an n-isogeny ϕ in dimension 1 into an N isogeny Φ in dimension $g = 2, 4, 8$. Taking N smooth and the N-torsion accessible (see Appendix A for this notion), we can decompose Φ into a product of small ℓ-isogenies, with $\ell = O(\log n)$. So for this choice of N, the HD representation is efficient according to our definitions.

But there will be a huge timing difference, depending on whether we can take $g = 2$ and $N = 2^e$ and work with rational points of 2^e-torsion, or if we need to work in dimension $g = 8$, with N having prime factors up to $O(\log n)$, and needing to work with torsion points defined over a field extension of degree up to $O(\log^2 n)$. For cryptographic applications, we want to find parameters allowing the former situation rather than the later. We will come back to this at several points during this survey.

We will explore several representations.

2.2. **The classical representations.** The *function representation* encodes E_1, E_2 by their short Weierstrass equations $E_i : y^2 = x^3 + a_i x + b_i$ (assume $p > 3$ here), and ϕ as a rational function $\phi(x, y) = \left(\frac{g(x)}{h(x)}, cy\left(\frac{g(x)}{h(x)}\right)'\right)$, where g, h are polynomials of degree $\leq n$ in x. This function representation takes linear space $O(n \log q)$ to encode, and linear time $O(n \log q')$ to evaluate.

The *kernel representation* encodes ϕ by the tuple $(E_1, \operatorname{Ker}\phi)$, this takes linear space. Vélu's formula can be used to convert from the kernel representation to the rational function representation (this includes computing a Weierstrass equation for E_2) in linear time. There is a slight ambiguity here since $\operatorname{Ker}\phi$ only determines ϕ up to post-composition by an automorphism of E_2. Since, unless $j(E_2) = 0, 1728$, the only automorphisms are multiplication by ± 1, we will ignore this subtlety.

A useful variant is to encode $\operatorname{Ker}\phi$ by a generator T, we call this the *generator representation*. In general, T may live in a field extension of $\mathbb{F}_q$, but when the generator is rational, this gives a compact (i.e., space efficient) representation of ϕ: $O(\log q)$ bits. Furthermore, in that case the isogeny image can be evaluated in $\widetilde{O}(\sqrt{n} \log q')$ by the `sqrtVelu` algorithm of [BDLS20]. A variant, if a generator T lives in a too large extension, is to use a *multigenerator representation*: $\operatorname{Ker}\phi = \langle T_1, \ldots, T_m \rangle$. For instance, if $n = \prod \ell_i^{e_i}$, we can take for T_i a generator of $\operatorname{Ker}\phi[\ell_i^{e_i}]$, by the CRT the T_i generate the full kernel, and may live in smaller field extensions than T.

Given a subgroup G of E_1, the question of whether we can find generators of G living in a small enough extension of $\mathbb{F}_q$ (by small we mean polynomial in $\log G$), will appear frequently in this paper. We say that the G-torsion is *accessible* when this is the case. For example, if $E[n]$ is accessible (for instance, we have $E[\ell_i^{e_i}] \subset E(\mathbb{F}_{q_i})$ with $\mathbb{F}_{q_i}$ a small field extension of $\mathbb{F}_q$ for each i), then all n-isogenies $\phi : E_1 \to E_2$ have a space efficient (multi) generator representation. We refer to Appendix A for more details.

An isogeny ϕ of degree $\leq n$ is completely determined by its image on $4n + 1$ distinct points [UJ18]. Let $G \subset E(\mathbb{F}_q)$ be a (rational) subgroup of order $N > 4n$. By additivity of ϕ, it suffices to know the images $\phi(P_i)$ of ϕ on generators $\langle P_i \rangle$ of G to know ϕ on G. We call this the *interpolation representation*. Indeed, one can use a standard rational function reconstruction to reconstruct the rational function $\frac{g(x)}{h(x)}$ from this interpolation data. If these generators are rational, this rational reconstruction can be done in quasi-linear time, using

standard algorithms [BCG+17]. The space needed for this representation depends on the fields of definition of the P_i, similarly to the generator representation of the kernel.

An alternative (when p is large enough compared to n), which is always space efficient, is to represent ϕ via the image $(\mathbf{P}, \phi(\mathbf{P}))$ of a formal point $\mathbf{P}$ at precision 2, and reconstruct ϕ (in quasi linear time) by solving a differential equation. We call this the *deformation representation*, because it encodes how ϕ deforms: see Section 3.3.2.

We can also use modular polynomials to represent an isogeny (this is the *modular representation*): if $(j(E_1), j(E_2))$ is a non singular point of the modular polynomial $\Phi_n(X, Y)$, this tuple is enough to reconstruct the deformation representation: one can use Φ_n to express the derivative $j'(E_2, dx/y)$ in term of $j'(E_1, dx/y)$ (for normalised differentials), which is enough to obtain the deformation of E_2 induced by the deformation of E_1 via the isogeny ϕ. Since $j(E_2)$ can be described as a specific root of $\Phi_n(j(E_1), Y)$ (via a deterministic ordering of roots), this representation is the only one, out of those presented in this paper, which, given E_1, only needs $O(\log n)$ extra bits of information to encode ϕ. However, the best algorithms currently know to evaluate $\Phi_n(j(E_1), Y)$ cost $\widetilde{O}(n^2 \log p + n \log q)$, so are not linear in n.

If $n = \prod \ell_i^{e_i}$, so that ϕ is an isogeny of smooth degree (say with smoothness bound $\ell_i \leq B = O(\log^C n)$), we can decompose $\phi : E_1 \to E_2$ as a product of small degree isogenies ϕ_i. Any reasonable representation of the small isogenies ϕ_i (say linear in their degree l_i) then gives an efficient representation of ϕ. We call this the *decomposition representation*.

Note however that given a representation of ϕ, computing its decomposition representation may be expensive; we will come back to this in Section 3.4. In the particular case when n is smooth and the n-torsion is accessible (this is always the case if n is powersmooth), then we can efficiently convert a (multi)generator representation into a decomposed representation, so the (multi)generator representation is efficient.

2.3. **The ideal representations.** Another type of representation, which is quite different from the ones above, is to represent the isogeny ϕ by an ideal I_ϕ.

There are two different cases here. When E_1, E_2 are supersingular curves defined over $\mathbb{F}_{p^2}$ (and such that all their geometric endomorphisms are already defined over $\mathbb{F}_{p^2}$), by Deuring's correspondence the isogeny ϕ is always represented by an ideal I_ϕ in a quaternion algebra. We call this the *supersingular ideal representation*.

The other case concerns oriented isogenies between oriented elliptic curves. Here, an orientation is an embedding of a quadratic imaginary order R inside $\mathrm{End}_k(E_1)$ and $\mathrm{End}_k(E_2)$. The isogeny ϕ is said to be oriented if $\phi \circ \gamma = \gamma \circ \phi$ for all $\gamma \in R$. The embedding of R inside $\mathrm{End}_k(E_1)$ is required to be primitive (or saturated), meaning that $\mathrm{End}_k(E_1) \cap (R \otimes_{\mathbb{Z}} \mathbb{Q}) = R$, but not the embedding of R inside $\mathrm{End}_k(E_2)$. If R is primitive in $\mathrm{End}_k(E_2)$, we say that ϕ is an horizontal oriented isogeny, otherwise we say that ϕ is an ascending oriented isogeny. In both cases, ϕ can also be represented by an ideal I_ϕ (this would not necessarily be true if we allowed isogenies of degree divisible by p), we call this the *oriented ideal representation*.

Whenever the Frobenius π_k is not trivial (not given by a multiplication by some integers), we can consider the natural $\mathbb{Z}[\pi_k]$ orientation on E_1, E_2; since ϕ is rational it will also be oriented. This case includes ordinary elliptic curves, or supersingular elliptic curves over $k = \mathbb{F}_p$. (Note however that $\mathbb{Z}[\pi_k]$ may not be saturated in $\mathrm{End}_k(E_1)$, so we may need to replace it by its saturation R to find an ideal representation of ϕ; such an ideal representation exists only if ϕ is horizontal or ascending.)

The ideal representation is always space efficient; it is even efficient in the supersingular or horizontal cases (provided we have an efficient representation of the orientation), but this is much harder to prove (see Section 4). To convert an ideal I of norm $N(I)$ (we define

$N(I)$ to be be the reduced norm in the supersingular case) into an isogeny, one can compute $E[I] \subset E[N(I)]$, the intersection of the kernel of all endomorphisms $\alpha \in I$. The isogeny ϕ_I corresponding to I is the isogeny with kernel $E[I]$, so this allows to go from the ideal representation to the kernel (or generator) representation in polynomial time in $n = N(I)$ (see Section 4 and Example A.5 for more details). A strategy to find an evaluation algorithm polylogarithmic in n is to find an equivalent ideal J of (power)smooth norm, and compute a decomposed representation of the isogeny ϕ_J (from which we can derive an algorithm to evaluate ϕ_I efficiently if we have an efficient representation of the endomorphism α linking J to I). Such a *smoothening* algorithm in the supersingular case is given by the KLPT algorithm [KLPT14] (a non heuristic version is proved under GRH in [Wes22]). For the oriented case, there is no known polynomial time smoothening algorithm. Hence the question of efficiently converting an arbitrary ideal I into an isogeny ϕ_I had remained an important open problem in isogeny based cryptography. Thus an efficient evaluation algorithm for an (horizontal) ideal representation, was only recently given in [PR23b], and makes essential use of the HD representation which will be the main topic of this survey.

The ideal representation is very convenient, but it has a crucial drawback: in the supersingular case, publishing I leaks the endomorphism ring of E_1 and E_2, which is the one thing that is supposed to stay secret for isogeny based cryptography. In [Ler22a], Leroux introduced the suborder representation which leaked less informations; this representation is now essentially superseded by the HD representation.

2.4. **The HD representation.** As explained in Section 1, the main byproduct of the SIDH attacks is that for any $N \geq n$ (coprime to p), by combining a lemma due to Kani in [Kan97] and Zahrin's trick, it is always possible to embed ϕ into a higher N-dimensional isogeny Φ between abelian varieties A, B of dimension $g = 2$ or 4 or 8. Taking N to be (power)smooth, and computing a decomposition representation of Φ, this allows to give an efficient representation of ϕ.

In practice, if N is prime to n, it is enough to know the action of ϕ on $E_1[N]$ to be able to reconstruct $\operatorname{Ker}\Phi$. From this kernel, one can use a higher dimensional version of Vélu's formula [Rob21; LR22] to compute the decomposition of Φ. We remark that, in some cases, there is an optimisation where we can relax the condition $N > n$ to $N^2 > n$, see [Rob23b, § 6.4], [DLRW24, Appendix C.2] or Appendix B. There is also a more recent version when we just need to know ϕ on a large enough subgroup rather than on the full N-torsion [CDM+24]. To simplify the exposition in this survey, we will mainly stick to the case $N > n$.

To describe the action of ϕ on $E_1[N]$, it suffices to give its action on a basis (P_1, P_2). It can be convenient to replace this basis by generators, like we did for the generator representation of the kernel: if $N = \prod l_i^{e_i}$ we give the action of ϕ on a basis of each $E_1[\ell_i^{e_i}]$, this can allow to work with smaller field extensions.

Technically, there are two subvariants of the HD representation, depending on how we represent Φ (see Definition 5.21): the one where we give the action of ϕ on (generators of) $E_1[N]$, which is essentially the kernel generator representation of Φ; and the one where we represent Φ by its decomposition into a product of smaller isogenies. The key point is that if N is smooth, and the basis of each $E_1[\ell_i^{e_i}]$ live in a small enough extension (which is always the case if N is powersmooth), then going from the kernel representation of Φ to its decomposition representation can be done efficiently, and then evaluating Φ (hence ϕ) amount to evaluating small ℓ_i-isogenies (in higher dimension), which is efficient.

We remark that the HD representation (in its kernel version) is very close to the interpolation representation of ϕ: we represent ϕ by its images $(P_{1,i}, P_{2,i}, \phi(P_{1,i}), \phi(P_{2,i}))$ on the basis $(P_{1,i}, P_{2,i})$ of the $E_1[\ell_i^{e_i}]$ torsion (say with ℓ_i prime to n to simplify), such that

$\prod \ell_i^{e_i} > n$. The key point is that, to evaluate ϕ, rather than reconstruct the function representation via a generic interpolation algorithm, like we used in the interpolation representation, there is a much more clever algorithm that uses the higher dimensional isogeny Φ to evaluate $\phi(P)$ in time polynomial in $\log n$, ℓ_i, $\log q'$ and the $\log q_i$, where $P_{1,i}, P_{2,i} \in E(\mathbb{F}_{q_i})$. (So not all HD representations are efficient, but we can always find an efficient HD representation by selecting the $\ell_i^{e_i}$ appropriately!) This is probably one of the most complicated structured[5] univariate interpolation algorithm in the world, but it works very well in practice!

We can summarize the HD representation as follows: if we know how to evaluate ϕ on enough nice points, we know how to efficiently evaluate it everywhere. Restated like that, it seems we have a bootstrapping problem: how do we find the evaluation of ϕ on these nice points in the first place? This was the main problem stated in [Rob22a] when this representation was introduced.

One answer is when we have a supersingular ideal representation I_ϕ of ϕ: we want to publish an efficient representation of ϕ without publishing I_ϕ which would leak the endomorphism ring. We can use the HD representation for that. This is the main idea behind SQIsignHD [DLRW24].

2.5. **Algorithms for the HD representation.** More generally, since the publication of [Rob22a], many algorithmic tools have been found to work directly with an efficient representation of an isogeny rather than the more customary kernel (or ideal) representation. See for instance [Rob22b; Ler23b; CLP24; NO23; PR23b]. The main goal of this survey is to give a convenient reference for these algorithms.

In the following, by efficient we still mean polynomial in $\log n$ and $\log q$.

(1) **Universality:** The HD representation is universal[6]: from any efficient representation of ϕ, we can efficiently extract an (efficient) HD representation. This means that an algorithm available for an HD representation applies to any efficient representation of ϕ, by converting it to an HD representation first if necessary. Notably, since the HD representation can be used to evaluate ϕ on a point P defined over a k-algebra, but we only need to be able to evaluate ϕ on points defined over finite field extensions of k to derive an HD representation, we see that any representation able to evaluate ϕ over fields k'/k can be converted to a representation able to evaluate it over any k-algebra!

(2) **Equality:** Given efficient representations of two isogenies $\phi_1, \phi_2 : E_1 \to E_2$, one can efficiently test if $\phi_1 = \phi_2$.

(3) **Composition and addition:** Given efficient representations of $\phi_1 : E_1 \to E_2$ and $\phi_2 : E_2 \to E_3$ (resp. $\phi_2 : E_1 \to E_2$), one can find[7] an efficient representation of $\phi_2 \circ \phi_1 : E_1 \to E_3$ (resp. $\phi_1 + \phi_2$).

(4) **Dual isogeny:** Given an efficient representation of $\phi : E_1 \to E_2$, one can find[7] an efficient representation of the dual isogeny $\widetilde{\phi} : E_2 \to E_1$.

(5) **Division:** Given efficient representations of $\phi : E_1 \to E_2$ and $\psi : E_1 \to E_1'$, one can find[7] an efficient representation of an isogeny $\phi' : E_1' \to E_2$ such that $\phi = \phi' \circ \psi$ if it exists, or answer $\perp$ if such an isogeny ϕ' does not exist. Likewise when $\psi : E_2' \to E_2$ and we try to write $\phi = \psi \circ \phi'$. As a particular case, when $\psi = [m]$, we can efficiently test if ϕ is divisible by m (equivalently: if $\operatorname{Ker}\phi \supset E[m]$), and if so compute an efficient representation of ϕ/m.

[5] Here the structure that is being exploited is that ϕ commutes with the addition law, and that the interpolation points have smooth order.

[6] This term was coined by Benjamin Wesolowski

[7] Efficiently!

(6) **Lifts and deformations**: Given an efficient representation of $\phi : E_1 \to E_2$ over a finite field $\mathbb{F}_q$, and given $R = \mathbb{F}_q[\varepsilon]/\varepsilon^m$ or $R = \mathbb{Z}_q/p^m\mathbb{Z}_q$, one can find[7] an efficient representation of the isogeny ϕ deformed/lifted to R.
(7) **Splitting isogenies:** If ϕ is an isogeny of degree $n = n_1n_2$ with $n_1 \wedge n_2 = 1$, ϕ decomposes uniquely as $\phi = \phi_2 \circ \phi_1$ where ϕ_i is of degree n_i. If we have an efficient representation of ϕ, we can find[7] an efficient representation of ϕ_2 and ϕ_1.
(8) **Pushforward of isogenies:** if $\phi_1 : E_1 \to E_2$ is of degree n_1, $\phi_2 : E_1 \to E_2'$ is of degree n_2, with $n_1 \wedge n_2 = 1$, and we have an efficient representation of ϕ_1, ϕ_2, then we can find[7] an efficient representation of the pushforward isogeny $\phi : E_1 \to E_3$ (this is the isogeny with kernel $\operatorname{Ker}\phi = \operatorname{Ker}\phi_1 + \operatorname{Ker}\phi_2$). We can write $\phi = \phi_2' \circ \phi_1 = \phi_1' \circ \phi_2$ where ϕ_2' (resp. ϕ_1') is the pushforward of ϕ_2 (resp. ϕ_1) by ϕ_1 (resp. ϕ_2), and, by the splitting algorithm above, we can find[7] an efficient representation of ϕ_1', ϕ_2'.
(9) **Kernel:** Given an efficient representation of an n-isogeny $\phi : E_1 \to E_2$, we can recover an equation for its kernel in quasi-linear time $\widetilde{O}(n)$ arithmetic operations. (Note that the output is of size $O(n \log q)$ so we cannot do better than linear time). If $E_2[n]$ is accessible, we can also find a generator representation of $\operatorname{Ker}\phi$ in poly-logarithmic time.

We note the following remarkable feature of the splitting and pushforward algorithms. By assumption, an isogeny representation of $\phi : E_1 \to E_2$ contains both the domain and codomain data. But in some of the representations we have discussed, only the domain is needed; the codomain can be recovered from the data. This is the case for the kernel representations (via the kernel equation or generators), and also for the ideal representations. We remark that, as long as we know how to evaluate ϕ (say in Weierstrass coordinates), we can evaluate ϕ on some points and recover the Weierstrass equation of E_2 via linear algebra, or we can evaluate ϕ on the formal group at low precision (see Section 3.2.1).

The HD representation of $\phi : E_1 \to E_2$ uses interpolation data $(P, Q, \phi(P), \phi(Q))$ for a basis (P, Q) of the N-torsion, hence in particular require to know the codomain E_2 already. However, when $n = n_1n_2$ with n_1 coprime to n_2, then the splitting algorithm recovers the middle curve $E_1 \to^{\phi_1} E_{12} \to^{\phi_2} E_2$ where $\phi = \phi_2 \circ \phi_1$ and ϕ_i is a n_i-isogeny. This is a powerful tool to construct new elliptic curves (see Examples 6.12 and 6.13), and it is amazing that we can extract E_{12} efficiently just from the interpolation data above, which seems on first sight to be completely unrelated to this curve!

In particular, if ϕ admits an efficient representation, then (ϕ, n_1) gives an example of an efficient representation of ϕ_1 where the codomain E_{12} of ϕ_1 is not specified in advance. For instance to construct such a ϕ, we can try to find $\phi_2 : E_{12} \to E_1$, of degree coprime to $\deg\phi_1$, so that ϕ is an endomorphism (we often have many convenient ways to represent endomorphisms, see Example 6.7).

3. The standard representations

In this section, we survey the classical representations: the function representation in Section 3.1, the kernel representations in Section 3.2, the interpolation representations and its variants in Section 3.3. An important case is when an isogeny can be decomposed into a product of isogenies of smaller degree, this is treated in Section 3.4.

3.1. The function representation.

Let $\phi : E_1 \to E_2$ be a cyclic isogeny of degree n over $k = \mathbb{F}_q$, with $E_i : y_i^2 = x_i^3 + a_ix_i + b_i$ given by short Weierstrass equations, and n prime to p. We can write ϕ as $\phi(x_1, y_1) = (R_1(x_1, y_1), R_2(x_1, y_1))$ for some rational functions

$R_1, R_2 \in k(E_1)$: we have a morphism $\phi^* : k(E_2) \to k(E_1)$ and R_1 (resp. R_2) is the image of x_2 (resp. y_2) by this morphism.

Since the divisor (0_{E_1}) has self intersection 1, and $\deg x_i = 2$, $\deg y_i = 3$, the rational functions R_1, R_2 have total degree $2n$ and $3n$ respectively (taking into account the point at infinity).

Since $\phi(-P) = \phi(P)$, we see that we can rewrite $\phi(x_1, y_1) = (R_1(x_1), y_1 R_2(x_1))$ where now $R_1(x_1), R_2(x_1) \in k(x_1) = k(\mathbb{P}^1)$.

Let $\omega_i = dx_i/y_i$ be the "canonical" basis of global differentials on E_i. We have $\phi^*\omega_2 = \frac{1}{c}\omega_1$ for some constant $c \neq 0$, because the space of global differentials is of dimension 1 (an elliptic curve has genus 1), and we only deal with separable isogenies so the action on differentials is non trivial. Plugging the formulas for ϕ, we obtain $\frac{dR_1}{y_1 R_2} = \frac{1}{c}\frac{dx_1}{y_1}$, so we can further simplify the formulas for ϕ to: $\phi(x, y) = (R(x), cyR'(x))$.

We say that ϕ is *normalised* if $c = 1$. Using the change of variable $x_2' = \frac{1}{c^2}x_2$ and $y_2' = \frac{1}{c^3}y_2$, we have $dx_2'/y_2' = cdx_2/y_2$, so $\phi : E_1 \to E_2'$ becomes normalised where $E_2' : {y_2'}^2 = {x_2'}^3 + a_2c^4x_2' + c^6b_2$.

The kernel of ϕ is given, along with 0_{E_1}, by all points that are sent to infinity by ϕ, hence by the denominator $h(x) = 0$ of $R(x) = \frac{g(x)}{h(x)}$. Let $K = \operatorname{Ker}\phi$, we have (since ϕ is separable the geometric points in the kernel have multiplicity one): $h(x) = \prod_{T \in K \setminus \{0\}}(x - x(T))$. We remark that if n is odd, $h(x)$ is a square because $T, -T$ are two distinct points with the same x coordinate.

The function representation takes space $O(n \log q)$, and $O(n \log q')$ to evaluate on a point $P \in E(\mathbb{F}_{q'})$.

3.2. **The kernel representations.**

3.2.1. *Kernel equation.* Conversely, given the equation $h(x) = 0$ representing a (cyclic) kernel K, we can recover the function representation. Kohel in [Koh96] gives these explicit formulas, adapted from Vélu [Vél71]: for $f(x) = x^3 + a_1x + b_1$, and $\sigma_1 = \sum_{T \in K \setminus \{0\}} x(T)$ the trace of h, we have that

$$\frac{g}{h}(x) = nx - \sigma_1 - f'(x)\frac{h'}{h}(x) - 2f(x)\left(\frac{h'}{h}\right)'(x), \tag{1}$$

is the function representation for a normalised isogeny $\phi : E_1 \to E_2, (x, y) \mapsto (\frac{g}{h}(x), y\left(\frac{g}{h}\right)'(x))$ with kernel K. Here $E_2 : y^2 = x^3 + a_2x + b_2$ is given by (see for instance [Feo10, § 8.2]):

$$a_2 = a_1 - 5t, \quad b_2 = b_1 - 7w, \quad t = \sum_{T \in K \setminus \{0\}} f'(T), \quad u = \sum_{T \in K \setminus \{0\}} 2f(T) + x(T)f'(T).$$

The kernel representation takes space $O(n \log q)$. More precisely, it is given by a polynomial $h(x)$ in $\mathbb{F}_q[x]$ of degree $n - 1$, and if n is odd we can write $h(x) = h_1(x)^2$, where h_1 is a polynomial of degree $(n - 1)/2$. Converting to the function representation takes $O(n)$ arithmetic operations using Equation (1).

Vélu's formula from [Vél71] were originally given in term of the generator representation: if $K = \langle T \rangle$, Vélu argues that the functions

$$\begin{aligned} R_1(P) &:= \left(x_1(P) + \sum_{T \in K \setminus \{0\}} x_1(P+T) - x_1(T) \right), \\ R_2(P) &:= \left(y_1(P) + \sum_{T \in K \setminus \{0\}} y_1(P+T) - y_1(T) \right) \end{aligned} \tag{2}$$

are invariant by translation by $T \in K$ and have the correct polar divisors to be the pullback by ϕ of Weierstrass coordinates x_2, y_2 on $E_2 = E_1/K$.

To recover the equation of E_2, Vélu use the formal group law of the elliptic curve. More precisely, he fixes the uniformiser $z_1 = -x_1/y_1$ on E_1, and look at the development of x_1, y_1 along z_1. He then plugs the formula of R_1, R_2 to obtain their development along z_1, hence the development of $z_2 = -R_1/R_2$ in term of z_1. Inversing this to get z_1 in term of z_2, he recover the development of R_1, R_2 in term of z_2, from which he obtain the equation of E_2. Weierstrass coordinates are not unique; Vélu explains that he made the choice of normalisations on R_1, R_2 such that z_2 and z_1 coincide up to order 5: $z_2 = z_1 + O(z_1^5)$. In particular, the resulting isogeny is normalised.

3.2.2. *Kernel generator.* To evaluate a cyclic isogeny ϕ on a point P given a generator T of the kernel $K = \langle T \rangle$, one can use the formulas from Equation (2), this costs $O(n)$ arithmetic operations in the compositum field of the fields of definition of P and T. An alternative is to first convert to the kernel equation: $h(x) = \prod_{i=1}^{n-1} (x - x(i \cdot T))$, this uses $O(n)$ arithmetic operations in the field of definition $\mathbb{F}_{q^e}$ of T, and then to use Equation (1) to evaluate on P in $O(n)$ arithmetic operation in $\mathbb{F}_{q'}$.

When the field of definition $\mathbb{F}_{q^e}$ of T is small, the generator representation gives our first compact isogeny representation: representing T takes space $O(e \log q)$. In the worst case we have $e = \Theta(n)$, so we do not gain anything compared to the $O(n \log q)$ size of the kernel representation, but in the best case $e = 1$ and the isogeny takes $O(\log q)$ to represent. (Note that in this case, since T is rational we have $n \mid \#E(\mathbb{F}_q)$, so necessarily $n = O(q)$).

Another advantage of the generator representation is that we can use the `sqrtVelu` algorithm of [BDLS20], which requires $\widetilde{O}(\sqrt{n})$ arithmetic operations in the compositum field of T and P to evaluate $\phi(P)$.

As mentioned in Section 2, we can also use a multigenerator representation $K = \langle T_1, \dots, T_m \rangle$. This is particularly useful when $n = \prod \ell_i^{e_i}$ is powersmooth, in that case we can take T_i to be a generator of $K[\ell_i^{e_i}]$, which will live in an extension of degree at most $\ell_i^{e_i}$. If the powersmooth bound B is $B = O(\log^{O(1)} n)$, this gives a compact representation. By contrast, the generator $T = T_1 + \cdots + T_m$ live in an extension of degree the compositum of all these fields, which can only be bounded by n in the generic case.

More precisely, since $K = \langle T \rangle$ is cyclic of degree n, π_q acts by multiplication by $\lambda \in (\mathbb{Z}/n\mathbb{Z})^*$ on T, and the order e of λ gives the degree of the extension where T is defined, so in particular $e \mid \phi(n)$.

Recall that we have defined a torsion subgroup G of cardinal N to be accessible whenever we can find generators that live in a small (polynomial in $\log N$) extension of $\mathbb{F}_q$. From this definition, we see that the multigenerator representation of the kernel K is space efficient whenever the K-torsion is accessible.

Remark 3.1 (Radical isogenies)**.** Sometimes for applications, like for the CGL hash function [CLG09], we wish to iterate ℓ-isogenies: we want to compute a cyclic ℓ^n-isogeny. As we

will see in Section 3.4, the best case happens when we have a rational generator of the big ℓ^n-isogeny, because we can then evaluate it in $\widetilde{O}(n \log \ell)$. However, it often happens that only the ℓ-torsion is rational (or lives in a small extension), and that the ℓ^n-torsion would live in a too big extension. The standard solution is then to compute this ℓ-torsion, extract the first generator of order ℓ from it, compute the first isogeny $E_0 \to E_1$ of degree ℓ, and iterate, computing the ℓ-torsion on E_1 again (typically by sampling points and multiplying by the cofactor).

The idea of radical isogenies is to start with the first kernel $K_0 = \langle P_0 \rangle$ too, but to compute the next kernel $K_1 = \langle P_1 \rangle \subset E_1$ directly, and to iterate this construction. This idea was introduced in [CDV20], where the authors showed that P_1 could be computed from a choice of ℓ-th root of the self Tate pairing $e_{T,\ell}(P_0, P_0)$. The original motivation of the radical isogeny formulas were for the CSIDH cryptosystem, and they have been improved in [OM22; CDHV22; Pri24; Dec24]. The theory of multiradical isogenies (radical isogenies in higher dimension) was developed in [CD21; Rob23c]

3.3. **Interpolation representations.**

3.3.1. *Standard Lagrange interpolation.* By the Cauchy-Schwarz inequality,

$$\deg(\phi_1 + \phi_2) \leq \left(\sqrt{\deg \phi_1} + \sqrt{\deg \phi_2}\right)^2.$$

Hence if ϕ_1, ϕ_2 are two n-isogenies such that $\phi_1 \neq -\phi_2$, $\deg(\phi_1 - \phi_2) \leq 4n$. It follows that if ϕ_1, ϕ_2 coincide on at least $4n + 1$ points, they have to be equal.

The interpolation representation of ϕ is given by $(E_1, E_2, (P_i, \phi(P_i)))$ for a list of $N > 4n$ points P_i. By the above argument, this completely determines ϕ.

In practice, we can recover ϕ as follows from the interpolation data:

- We first recover interpolation data for $R(x) = \frac{g}{h}(x)$, with the notations from Section 3.1. Each interpolation data $(P_i, \phi(P_i))$ gives an interpolation data $(x(P_i), R(x(P_i)))$. Since the points P_i are distinct, and only $\pm P_i$ have the same x-coordinate, we obtain at least $2n + 1$ different interpolation points for R.
- The function $R(x)$ has total degree $2n$ seen as a function on E (taking into account the point at infinity), and since $\deg(x) = 2$ we get that $\deg_x R \leq n$. Hence our $2n + 1$ points of interpolation are enough to recover $R(x)$ in quasi-linear time $\widetilde{O}(n)$ by the standard rational function reconstruction algorithms [BCG+17].
- It remains to recover the normalisation constant c, which we can obtain from any point P_i such that $\phi(P_i)$ is not of 2-torsion. There are at most $4n$ such points, and we have $4n + 1$ points, so we can always find one.

The interpolation representation takes space $O(n \log q)$, assuming that all the P_i are defined in $\mathbb{F}_q$. On the other hand, if $P_i \in E_1(\mathbb{F}_{q^e})$, we can act by the Galois action (i.e., the Frobenius) on the interpolation data $(P_i, \phi(P_i))$ to obtain "for free" e different interpolation points.

But in fact, we can do better: by additivity of ϕ, knowing the interpolation data on some points P_i allows to recover the action of ϕ on the full subgroup $G = \langle P_i \rangle$. So it suffices to give interpolation data for generators P_i of a rational (for simplicity) subgroup $G \subset E_1$ to recover ϕ by interpolation, as long as $\#G > 4n$. Assume that $\#G = N$, with $N = \prod \ell_i^{e_i}$. Like in the multi generator representation for the kernel, it can be helpful to diminish the size of the representation to take generators of each of the CRT subgroups $G[\ell_i^{e_i}]$. (However, the complexity of the rational function reconstruction will depend on the degree of the field of definition of all points of G.)

Note that here N is decoupled from n. This can be used to find a space efficient representation of ϕ even if its kernel K has its torsion not accessible: we only need to find a large enough accessible torsion subgroup $G \subset E$.

For instance, if $P \in E_1(\mathbb{F}_q)$ is a rational point of N-torsion, we can use $(P, \phi(P))$ (along with (E_1, E_2, n) as usual) as a compact interpolation representation of any n-isogeny ϕ with $4n < N$. We remark that in this case, $N = O(q)$, hence $n = O(q)$.

3.3.2. *The deformation representation as an Hermite-Padé interpolation.* There is a convenient way to find such a rational torsion "point" of large order. Namely, let $\mathbf{P}$ be fat point above 0_{E_1}, that is a point $\operatorname{Spec} k[\varepsilon]/\varepsilon^2 \to E$ such that the natural composition $\operatorname{Spec} k \to \operatorname{Spec} k[\varepsilon]/\varepsilon^2 \to E$ gives 0_{E_1}. One can see $\mathbf{P}$ as the choice of a tangent vector $v \in T_{O_E}(E)$ of E at 0_E. Then $\mathbf{P}$ is a rational "point", that we can also view as a formal point in the formal group of E truncated to precision 2, which is of order[8] p, because $[n]$ acts by multiplication by n on the tangent space at 0_{E_1}.

Concretely, since ϕ maps the neutral point to the neutral point, to give the interpolation data $(\mathbf{P}, \phi(\mathbf{P}))$ (to precision 2) is equivalent to giving the action of ϕ on the tangent space of E_1 and E_2 at their neutral points, or equivalently to give the normalisation constant c such that $\phi^*\omega_2 = \frac{1}{c}\omega_1$. Indeed, if we use $z_1 = x_1/y_1$ as a uniformiser on E_1, then a formal point at precision m is given by $\mathbf{P} = a_1 z_1 + a_2 z_1^2 + a_3 z_1^3 + O(z_1^{m+1})$. We have explicit formulas for $x_1 = z_1^{-2} + \cdots$, $y_1 = z_1^{-3} + \cdots$, from the curve equation of E_1, so we have the Weierstrass formal coordinates of $\mathbf{P}$, to which we can apply ϕ to obtain the Weierstrass formal coordinates x_2, y_2 of $\phi(\mathbf{P})$. Letting $z_2 = x_2/y_2$, we can thus recover $z_2(\phi(\mathbf{P}))$ to precision m: $z_2(\phi(\mathbf{P})) = \frac{a_1}{c} z_1 + a_2' z_1^2 + \ldots + O(z_1^{m+1})$. In particular, $dz_2 \circ d\phi = \frac{1}{c} dz_1$. Since $dz_i = dx_i/y_i$, we see that working at precision $m = 2$ is enough to recover c such that $\phi^*\omega_2 = \frac{1}{c}\omega_1$.

By the same argument as in the usual interpolation reconstruction, the isogeny ϕ is completely determined from this Hermite-Padé interpolation data c as long as p is large enough compared to n: $p > 4n$. To reconstruct ϕ in practice, i.e. recover the rational function R, one can solve the differential equation

$$c^2(x^3 + a_1 x + b_1)R'(x)^2 = R(x)^3 + a_2 R(x) + b_2; \tag{3}$$

derived from the equation $\phi^*\omega_2 = \frac{1}{c}\omega_1$.

This differential equation can be solved in quasi-linear time $\widetilde{O}(n)$ arithmetic operations [BMSS08] using fast methods on power series, to first reconstruct R as a power series and then as a rational function, as long as $p > 8n - 5$.

When n is too large compared to p, one can instead take a lift $\widetilde{\phi} : \widetilde{E_1} \to \widetilde{E_2}$ of ϕ to $\mathbb{Z}_q/p^m\mathbb{Z}_q$ (where $\mathbb{Z}_q$ is the ring of Witt vectors over $\mathbb{F}_q$), at p-adic precision m large enough: $n = O(p^m)$. Such a technique is used in [LS08] to reconstruct an isogeny from roots of the modular polynomial ϕ_n even in cases where p is small compared to n.

We call this representation the *deformation representation*; the advantage of this representation is that it takes space $O(m \log q) = O(\log q + d \log n)$ for any n-isogeny, where $q = p^d$. The rational function reconstruction takes quasi-linear time. This representation will (one day) be explained in more details in [KR22], in the meanwhile see [Rob21, § 4.7.3].

[8] This is not a coincidence, as we will see in Remark 3.2, by [Oda69], the Dieudonné module $\mathbb{D}(E[p])$ of $E[p]$ is canonically isomorphic to the de Rham cohomology $H^1_{DR}(E)$, and the Frobenius filtration on $E[p]$ corresponds to the Hodge filtration on $H^1_{DR}(E)$ [Oda69, Corollary 5.11], so in particular $D(E[\pi_p]) \simeq H^0(E, \Omega_E)$ (up to a Frobenius twist). In other words, differentials on E corresponds via the Dieudonné anti-equivalence of category to infinitesimal points of p-torsion in $E[\pi_p]$.

The reason for the name *deformation representation* comes from the following useful reformulation. The Kodaira-Spencer isomorphism gives a canonical isomorphism between the Sym^2 of the tangent space of an elliptic curve E at 0_E and the tangent space of the moduli stack $\mathcal{A}_1$ parametrizing elliptic curves at E (see Appendix E). So, reformulating in terms of differential, the choice of a differential ω_E on E is essentially the same as the choice of a deformation $\widetilde{E}/\operatorname{Spec}(k[\varepsilon]/\varepsilon^2)$ of E (essentially because of the Sym^2, which means that $\pm\omega_E$ gives the same deformation). In terms of modular functions, this is just a reformulation of the fact that the modular function $j'(\tau)$ is of weight 2, where $j'(\tau) = dj(\tau)/2\pi i d\tau$. From the algebraic interpretation of modular forms as sections of suitable powers of the Hodge line bundle, a modular function of weight 2 such as j' means that j' depends (in a functorial way) on both the choice of E and a differential ω_E on E, and that $j'(E, \lambda\omega_E) = \lambda^2 j'(E, \omega_E)$.

All of this can be made very explicit: if $\omega_E = dx/y$ is the canonical differential form on $E : y^2 = x^3 + ax + b$ we have $j'(E, \omega_E) = 18j(E)b/a$ by [Sch95, § 7]. Conversely, fixing $j'(E, \omega_E)$ for some differentials ω_E on E fixes ω_E up to a sign because j' is of weight 2, hence in particular fixes a, b such that $\omega_E = dx/y$ on $E : y^2 = x^3 + ax + b$ because a is of weight 4 and b is of weight 6.

And the deformation $\widetilde{E}$ associated to (E, ω_E) is given by $\widetilde{E} : y^2 = x^3 + a(\varepsilon)x + b(\varepsilon)$ such that $j(\widetilde{E}) = j(E) + j'(E, \omega_E)\varepsilon$ modulo ε^2, which gives a linear equation between the coefficients a', b' of $a(\varepsilon) = a + a'\varepsilon$ and $b(\varepsilon) = b + b'\varepsilon$. We cannot do better because multiplying $a(\varepsilon)$ and $b(\varepsilon)$ by $u(\varepsilon)^4$ and $u(\varepsilon)^6$ respectively with $u(\varepsilon) = 1 + \gamma\varepsilon$ does not change the isomorphism class of $\widetilde{E}$ and the curve still reduces to E.

In other words, the following data are essentially equivalent for an elliptic curve E:

- A short Weierstrass equation for E, and its associated differential $\omega_E = dx/y$
- The j-invariant $j(E)$ and a choice of differential ω_E up to a sign
- $j(E)$ and $j'(E, \omega_E)$
- The j-invariant $j(\widetilde{E})$ of the deformation $\widetilde{E}$ of E to $k[\varepsilon]/\varepsilon^2$ associated to $\mathrm{Sym}^2\,\omega_E$

Now, given a deformation $\widetilde{E_1}$ of E_1, the isogeny $\phi : E_1 \to E_2$ deforms uniquely to an isogeny $\widetilde{E_1} \to \widetilde{E_2}$. If $\widetilde{E_1}$ is the deformation associated to a differential $\mathrm{Sym}^2\,\omega_{E_1}$, then $\widetilde{E_2}$ is the deformation associated to the differential $\mathrm{Sym}^2\,\omega_{E_2}/n$ where $\phi^*\omega_{E_2} = \omega_{E_1}$ (see [KPR24]). Given two short Weierstrass equations $y^2 = x^3 + a_i x + b_i$ for E_1, E_2, the normalisation constant $\phi^* dx_2/y_2 = \frac{1}{c} dx_1/y_1$ then shows that if $\widetilde{E_1}$ is the deformation associated to $\omega_{E_1} = dx_1/y_1$, then $\widetilde{E_2}$ is the deformation associated to $\omega_{E_2} = c dx_2/y_2$.

In summary, given E_1, E_2, the deformation representation consists of any of these essentially equivalent data (meaning that we can easily switch between them):

- The pullback action of ϕ on differentials;
- The datum $j'(E_1, \omega_{E_1})$, $j'(E_2, \omega_{E_2})$ for the modular function j', where $\omega_{E_1} = \phi^*\omega_{E_2}$;
- The isogeny normalisation constant;
- The deformation $\widetilde{\phi} : \widetilde{E_1} \to \widetilde{E_2}$ to any non trivial deformation $\widetilde{E_1}/(k[\varepsilon]/\varepsilon^2)$ of E_1;
- The action of ϕ on the formal group to precision $m = 2$;
- The differential equation Equation (3), from which we can recover ϕ in $\widetilde{O}(n)$ (if p is large enough).

Remark 3.2 (Dieudonné modules, points of p-torsion and differentials)**.** From the point of view of Dieudonné modules, the Dieudonné module $\mathbb{D}(E[p^n])$ (in the contravariant

Dieudonné theory) is precisely given by the de Rham cohomology $H^1_{DR}(\widetilde{E})$ of a lift $\widetilde{E}$ of E to p-adic precision n. This is a consequence of the fact that the Dieudonné module $\mathbb{D}(E(p))$ of the p-divisible group is isomorphic to the crystal $H^1_{\text{crys}}(E, \mathbb{Z}_p)$ associated to E, which itself can be computed as the hypercohomology of the de Rham Witt complex [Ill79]. Furthermore, by the Serre-Tate theorem lifts $\widetilde{E}$ of E corresponds to lifts of the p-divisible group $E(p)$, which themselves correspond by the Grothendieck-Mazur theorem to lifts of their associated crystals (which over a field is the Dieudonné module), which is encoded by the linear data of a choice of lift of their Hodge filtration from modulo p to modulo p^n. The filtration on $\mathbb{D}(E[p^n])$ modulo p^n associated to a choice of lift $\widetilde{E}$ is precisely given by the Hodge filtration on $H^1_{dR}(\widetilde{E}, \mathbb{Z}/p^n\mathbb{Z})$ via the isomorphism above.

This is really a beautiful theory, and unfortunately it is hard to find a concise reference for all these facts. At the core, Grothendieck's crystalline topology should be seen as a far reaching generalisation that an infinitesimal point can be associated to some differential data.

Also, a very useful fact proved by Tate in [Tat67], is that over a complete Noetherian local ring R with residue field of cararacteristic p, connected p-divisible groups correspond to formal Lie groups (when seen as fppf sheafs; and the Dieudonné module of the group is the same as the Cartier module of the Lie group). And as expected, the formal Lie group corresponding to the connected part of the p-divisible group $E(p)$ is precisely the formal group of E.

All these results extend to an abelian variety A. For the isogeny practitioner, what the Dieudonné theory means is that the p-torsion of A corresponds to $H^1_{DR}(A)$ (and a point of p^n-torsion to a De Rham differential of a lift of A modulo p^n). As mentioned earlier, by [Oda69, Corollary 5.11], this is refined by the Hodge decomposition $0 \to H^0(A, \Omega^1_{A/k}) \to H^1_{dR}(A) \to H^1(A, O_A) \to 0$ which corresponds via the Dieudonné functor to the Frobenius filtration $0 \to A[\hat{\pi}] = \Im\pi \to A[p] \to A[\pi] = \operatorname{Ker}\pi \to 0$; more precisely $\mathbb{D}(A[\hat{\pi}]) \simeq H^1(A, O_A) \simeq H^0(A^\vee, \Omega^1_{A^\vee/k})^\vee$ while $\mathbb{D}(A[\pi])$ is a Frobenius twist of $H^0(A, \Omega^1_{A/k})$. In particular, the choice of an infinitesimal point in $A[\pi]$ (note that this is always a connected infinitesimal proper group scheme) corresponds to a choice of global differential on A. And by the discussion above, the formal group law encodes the connected part $A^\circ(p)$ of the p-divisible group.

In some sense, since the global differentials form a vector space of dimension g for an abelian variety of dimension g (because it is smooth), we always have a "free" basis of rank g of points of p-torsion. Furthermore, to any algorithm which is expressed in term of usual geometric points on A, as long as it is functorial enough to work as well with points with multiplicities, so in particular p-torsion points, then there should exist a traduction of this algorithm (via Dieudonné's antiequivalence of category) which can work in terms of differentials. (This does not mean that the resulting algorithm is still as efficient though). This is precisely what we have done above: we had an algorithm to interpolate an n-isogeny from its value at a point of ℓ-torsion, with $\ell \gg n$, and we adapted it to an algorithm that reconstitutes the isogeny from how it acts on the global differential dx/y, as long as $p \gg n$. The main advantage, as we have noted, is that the global differential is always there, while the point of ℓ-torsion may live in a large extension. This point of view can often be helpful in isogeny based cryptography when one need extra points of ℓ-torsion and how isogenies act on them: using $\ell = p$ allows to work with rational "points".

The last remark is that, implicitly, many existing isogeny algorithms in the literature track these points of p-torsion (i.e. track the global differentials). It is well known that Vélu formula gives normalised isogenies; and the condition $\varphi^*\omega_2 = \omega_1$ for differentials / p-torsion "points"

is the pendant of the equation $\varphi(P_1) = P_2$ for normal (étale) points. It is a bit less known, but still true, that the usual isogeny formulas in Mongomery coordinates or theta coordinates still track differentials. For Vélu formulas, the differential was encoded through a choice of short Weierstrass equation $y^2 = x^3+ax+b$, and indeed the Weierstrass coefficients a, b are modular forms of weight 4 and 6 respectively. In the Montgomery model $y^2 = x^3 + \mathcal{A}x^2 + x$, the coefficient $\mathcal{A}$ is only a modular function (of level $\Gamma^0(4)$), so does not keep track of differentials. Likewise, the level 2 projective theta null point $(\theta_0(E) : \theta_1(E)) = \theta_1(E)/\theta_0(E)$ is given by a modular function (of level $\Gamma(2,4)$). But in practice, during the isogeny algorithms we work with the numerator A and denominator C of $\mathcal{A} = A/C$ separately to avoid divisions, and likewise for the numerator $\theta_1(E)$, and denominator $\theta_0(E)$ of the theta null point. This time we do have modular forms. We refer to [KNRR21, § 4.3] for more details on the modular interpretation of the theta isogeny formulas, to [KNRR21, § 4.5] for applications on how to compute Siegel modular forms algebraically, and to [Rob21, § 6.4] for applications to point counting via canonical lifts.

3.3.3. *Modular polynomials.* This point of view allows us to make the link with the *modular representation*. Recall that the modular polynomial $\Phi_n(X,Y)$ is a symmetric polynomial in $\mathbb{Z}[X,Y]$, such that $\Phi_n(j(E_1),Y) = \prod(Y - j(E_{1,i}))$ where $E_{1,i}$ are all the n-isogeneous codomain curves from E_1. When this evaluated polynomial has no multiplicity, a root $y = j(E_2)$ completely determines the n-isogeny $\phi : E_1 \to E_2$.

Elkies algorithm [Elk92; Elk97] gives a way to reconstruct ϕ in practice from $\Phi_n, j(E_1), j(E_2)$. We can reformulate it as follows: start with a short Weierstrass equation for E_1, and look at the deformation $\widetilde{E_1}$, with $j(\widetilde{E_1}) = j(E_1) + \varepsilon j'(E_1, \omega_{E_1})$ associated with $\mathrm{Sym}^2\,\omega_{E_1}$ for the differential $\omega_{E_1} = dx_1/y_1$. Then ϕ deforms to $\widetilde{\phi} : \widetilde{E_1} \to \widetilde{E_2}$, where $\widetilde{E_2}$ is associated to the differential $\mathrm{Sym}^2\,\omega_{E_2}/n$, where $\phi^*\omega_{E_2} = \omega_{E_1}$. In other words, $j(\widetilde{E_2}) = j(E_2) + \varepsilon j'(E_2, \omega_{E_2}/\sqrt{n}) = j(E_2) + \varepsilon j'(E_2, \omega_{E_2})/n$. Plugging the equation $\Phi_n(j(\widetilde{E_1}), j(\widetilde{E_2})) = 0$ amount to differentiating Φ_n, and gives the equation $j'(E_1, \omega_{E_1})\partial\Phi_n/\partial X(j(E_1), j(E_2)) + j'(E_2, \omega_{E_2})/n\partial\Phi_n/\partial Y(j(E_1), j(E_2)) = 0$. This allow us to recover $j'(E_2, \omega_{E_2})$ and also $j(\widetilde{E_2}) = j(E_2) + \varepsilon j'(E_2, \omega_{E_2})/n$. From $j'(E_2, \omega_{E_2})$, we recover the short Weierstrass equation of E_2 such that the isogeny ϕ is normalised with respect to ω_{E_1} and $\omega_{E_2} = dx_2/y_2$. In other word, we recover the deformation representation of ϕ (normalised in this case to have $c = 1$), from which we can recover ϕ by using [BMSS08] in quasi-linear time when $n = O(p)$. See [Rob21, § 5.4.1] for more details, and [KPR24] for a generalisation to abelian surfaces.

Given E_1, in the case where the codomains of all the n-isogenies from E_1 are different, ϕ is completely determined from $(E_1, j(E_2), n)$ In fact, we could even replace $j(E_2)$ by using a deterministic numerotation of the roots of $\Phi_n(j(E_1), Y)$, and giving which root number corresponds to $j(E_2)$; this representation takes $O(\log n)$ space rather than $O(\log p)$.

On the other hand, to reconstruct the isogeny ϕ from this data, we need to compute Φ_n, which takes quasi-linear time in its size $\widetilde{O}(n^3)$ [Eng09; BLS12; KR24]: this time the reconstruction is not quasi-linear in n. But for the reconstruction we just need the evaluations $\Phi_n(j(E_1), Y)$ and $\partial\Phi_n/\partial X(j(E_1), Y)$ which can be computed faster [Sut13b; Kie20; Ler23a; Rob22b]. For instance, we have algorithms in $\widetilde{O}(n^2 \log p)$ to recover Φ_n modulo p directly. However, we have no algorithms so far able to compute $\Phi_n(j(E_1), Y)$ in quasi-linear time $\widetilde{O}(n \log q)$.

An alternative approach to reconstruct ϕ given (E_1, E_2, n), without using Φ_n, is given in [DHPS16].

3.4. **Decomposing an isogeny.** So far, we have seen some representations that were space efficient, but none able to evaluate ϕ on a point P in better than linear time. When $n = \prod \ell_i^{e_i}$ is smooth, the decomposition representation will be our first efficient representation. The key point is that we can decompose ϕ as a product of ℓ_i-isogenies: $\phi = \phi_1 \circ \phi_2 \circ \dots$.

To represent ϕ, it then suffices to give a representation of each of the intermediate ϕ_i; and any decent representation of ϕ_i (for instance the kernel representation or even the function representation), where by decent we mean a representation which takes linear space and time in the degree, will then give a representation of ϕ which takes space $O(B \log n \log q)$ and time $O(B \log n \log q')$ for evaluation, where B is a smoothness bound on n. More precisely, the *decomposed representation* takes space $O((\sum e_i \ell_i) \log q)$ and can be evaluated in $O((\sum e_i \ell_i) \log q')$. If n is smooth (B is polynomial in $\log n$), we obtain our fist representation that is both compact and efficient.

One can also look at the cost of converting from one of the above representations to the decomposed representation. The main case of interest is the generator representation: $K = \langle T \rangle$. For simplicity, we will focus on the particular case where $n = 2^e$ and $T \in E_1(\mathbb{F}_q)$ is a point of 2^e-torsion.

The naive way to decompose ϕ as a product of 2-isogenies is to start with T, compute $T_1 = 2^{e-1}T$ of order 2 which generates the kernel of our first 2-isogeny ϕ_1, compute the equations for ϕ_1 using Vélu's formulas from Section 3.2, and then pushforward T via ϕ_1 to get $\phi_1(T)$ which generates a kernel of order 2^{e-1} on $E_1/\langle T_1 \rangle$, and iterate this construction. This costs $O(e^2)$ arithmetic operations because we need $O(e^2)$ doublings.

A clever solution to achieve a better complexity was introduced in [DJP14]: the authors show how one can push more (carefully chosen) points along our intermediate isogenies ϕ_i to obtain a complexity of $O(e \log e)$ arithmetic operations to obtain the decomposition of ϕ.

The same method holds in the general case: provided that n is smooth and the kernel torsion is accessible (e.g., n is powersmooth, or $n = 2^e$ and T lives in a small extension), one can convert a multigenerator representation of the kernel into a decomposed representation in polylogarithmic time.

More precisely (restricting to ϕ cyclic for simplicity, but the same bounds hold in the general case):

Proposition 3.3. *Let $n = \prod_{i=1}^{m} \ell_i^{e_i}$, $T_1, \dots, T_i, \dots, T_m$ generators of $K[\ell_i^{e_i}]$, where $T_i \in E(\mathbb{F}_{q^{d_i}})$. Then one can compute the decomposed representation of the isogeny ϕ with kernel K in time $\widetilde{O}(\sum_i e_i \ell_i \log q (d_i \log e_i + \sum_{j>i} d_j)) = \widetilde{O}(m^2 d e \ell \log q)$ where d (resp. e, ℓ) is a bound on the e_i (resp. d_i, ℓ_i).*

An alternative method, using `sqrtVelu`*, cost $\widetilde{O}(\sum_i e_i \sqrt{\ell_i} \log q (d_i + \sum_{j>i} (d_i \vee d_j)) = \widetilde{O}(m^2 d' e \sqrt{\ell} \log q)$ where this time d' is the maximum of the degree of the compositum field extension of the fields of definitions of T_i, T_j for i, j.*

Proof. We refer to Remark A.2 for our assumptions about the lattice of field extensions used to represent our accessible CRT basis $\langle T_i \rangle$.

One first decomposes the isogeny with kernel $K[\ell_1^{e_1}]$ using its generator T_1 in $O(e_1 \log e_1 \ell_1)$ arithmetic operations over $\mathbb{F}_{q^{d_1}}$; this costs $\widetilde{O}(e_1 \log e_1 \ell_1 d_1 \log q)$. Then we push the other generators $T_2, \dots, T_m$ and we iterate. Pushing T_i through this $\ell_1^{e_1}$ isogeny takes $O(e_1 \ell_1)$ arithmetic operations over $\mathbb{F}_{q^{d_i}}$, so time $\widetilde{O}(e_1 \ell_1 d_i \log q)$, using the kernel representation.

An alternative to using the kernel representation to represent the intermediate ℓ_i-isogenies $\phi_{i,u}$ is to keep the generator representation and use the `sqrtVelu` formulas. This replaces the

$O(\ell_i)$ part by $\widetilde{O}(\sqrt{\ell_i})$, but on the other hand one has to work with the compositum field of the field of definition of the generator T_i' of the kernel of ϕ_i (which is included in the field of definition of T_i), and the field of definition of the current image T_j' of T_j (which is included in the field of definition of T_j). This compositum is thus of degree at most $d_i \vee d_j$. □

In isogeny based cryptography, for *practical* reasons we prefer to work with isogenies of smooth degree defined by rational generators, this is in practice much faster than working with general accessible torsion, which requires (small) field extensions. Still, it can be convenient to take small field extensions for some applications, and the case of non rational generators has been quite optimised too, see [EPSV; BGDS23]

Example 3.4 (Decomposing an isogeny from its kernel equation)**.** If $\phi : E_1 \to E_2$ is given by its kernel equation $K = \operatorname{Ker}\phi$: $H(x) = 0$, we can apply Proposition 3.3 by treating the point $\mathbf{P} : x \mod H$ as a "formal" generator of K (working in x-only coordinates for simplicity).

Let's assume that $n = n_1 n_2$. We can first compute $n_2\mathbf{P}$, then compute its characteristic polynomial (for instance via a resultant). To avoid divisions by potentially non invertible elements modulo H, one can work with the homogenisation $H(X, Z)$ of $H(x)$, and work in $(X : Z)$ coordinates; the homogenised characteristic polynomial of $n_2\mathbf{P}$ is given by $H_1(X, Z) = \prod_{P\in K}(Z(n_2P)X - X(n_1P)Z) = \prod_{Q\in K[n_1]}(Z(Q)X - X(Q)Z)^{n_2}$. Taking the gcd with H, we recover $G_1(X, Z) = \prod_{Q\in K[n_1]}(Z(Q)X - X(Q)Z)$, where G_1 gives the equation of the kernel $K_1 = K[n_1]$.

We remark that non invertible elements modulo H are actually useful since taking a gcd with H allows to partially factorize it. For instance we could recover G_1 directly (without a resultant) by computing $n_1\mathbf{P} = (P(X, Z), Q(X, Z))$ where Q is the division polynomial $\phi_n(X, Z)$ computed modulo H, and then taking $G_1 = Q \wedge H$. This is essentially the same as computing $\phi_n \mod H$ via the recurrence formula for ϕ_n where we reduce modulo H at each step, and this costs $O(\log n)$ arithmetic operations in $k[x]/H(x)$.

From this equation for K_1, we can compute the isogeny equation for the first step $\phi_1 : E_1 \to E_1'$ of the isogeny: $\phi = \phi_2 \circ \phi_1$. We can then compute $\mathbf{P}_2 = \phi_1(\mathbf{P})$, and take a resultant (or better use power projection [Sho99]) to get the kernel equation $H_2(X, Z) = 0$ of $\operatorname{Ker}\phi_2$. Using [KU11] for the power projection over a finite algebra, this step can be done in pseudo-linear time in n.

We refer to [EPSV; BGDS23] for optimisations, like using minimal polynomials rather than characteristic polynomials or using the Frobenius action.

4. The ideal representations

In this section we survey the ideal representation. We explain the link between ideals and isogenies in Section 4.1, then we treat the case of supersingular curves in Section 4.2 and the case of oriented curves (in particular ordinary curves) in Section 4.3.

4.1. **Isogenies represented by ideals.**

4.1.1. *Ordinary isogeny graph: volcanoes.* It is well known that if $E/\mathbb{F}_q$ is an ordinary elliptic curve, its ℓ-isogeny graph form the structure of a volcano, see [Koh96; FM02; Sut13a].

Let $\phi : E_1 \to E_2$ be an n-isogeny between elliptic curves defined over a finite field $\mathbb{F}_q$. In the ordinary case, we say that ϕ is horizontal if $\operatorname{End}(E_1) = \operatorname{End}(E_2)$ (in the ordinary case $\operatorname{End}_{\overline{\mathbb{F}}_q}(E) = \operatorname{End}_{\mathbb{F}_q}(E)$ is a quadratic imaginary field, so the notation is unambiguous), ascending if $\operatorname{End}(E_1) \subset \operatorname{End}(E_2)$ (and it is not horizontal), and descending if $\operatorname{End}(E_1) \supset \operatorname{End}(E_2)$ (and it is not horizontal). If n is not prime we can also have incomparable orders.

An horizontal or ascending isogeny ϕ of kernel K can always be represented by an ideal $I \subset \mathrm{End}(E_1)$: if we let $I = I(\phi) := \{\alpha \in \mathrm{End}(E_1), \alpha(K) = 0\}$ to be the set of all endomorphisms that are 0 on $K = \mathrm{Ker}\,\varphi$, then we have $E[I] = K$, where $E[I] := \{P \in E(\overline{\mathbb{F}}_q) \mid \alpha(P) = 0 \forall \alpha \in I\}$ (we have the obvious inclusion $K \subset E[I]$ by definition, and one can prove the converse when ϕ is not descending). Note that $E[I] \subset E[N(I)]$. In fact, I can always be represented as $I = (\alpha, N(I))$, and $E[I] = E[\alpha] \cap E[N(I)]$. The isogeny ϕ is horizontal if and only if I is invertible in $\mathrm{End}(E_1)$, otherwise ϕ is ascending and $\mathrm{End}(E_2) = O(I)$ is the order associated to I. Note that if ϕ is descending, it cannot be represented by an ideal in $\mathrm{End}(E_1)$, but its dual $\widetilde{\phi} : E_2 \to E_1$ can be represented by a non invertible ideal in $\mathrm{End}(E_2)$. For more details, see [Wat69; Koh96; Kan11].

In particular, if $R = \mathrm{End}(E_1)$, we have a commutative group action (which is free and transitive) from the Picard group $\mathrm{Pic}(R)$ of R (the class group of invertible ideals, this is the same as the usual class group when R is a maximal quadratic order) on the set of elliptic curves E' horizontal isogeneous to E_1 (meaning that $\mathrm{End}(E') = R$). To an invertible ideal class $[I]$ with representative I, we associate the codomain E'_I of $\phi_I : E' \to E'_I$. This group action was first used in the context of cryptography by Couveignes in 1997 (but published much later in [Cou06]), and revisited in [RS06; DKS18]. The first practical version was CSIDH [CLMPR18]. Having a commutative group action allows to translate many cryptographic constructions coming from the DLP problem. For instance the CSIDH key exchange is very similar to the Diffie-Helman's key exchange. However, while Shor's quantum polynomial time algorithm for the DLP does not apply to group actions, there exists a quantum subexponential time algorithm due to Kuperberg [Kup05; Pei20].

4.1.2. *Supersingular isogeny graphs.* If E_1, E_2 are supersingular elliptic curves defined over $\mathbb{F}_{p^2}$, with all their geometric endomorphisms defined over $\mathbb{F}_{p^2}$ (equivalently $\mathrm{End}_{\mathbb{F}_{p^2}}(E_i)$ is of rank 4), then ϕ is "almost horizontal", in the sense that $\mathrm{End}(E_i)$ are both maximal quaternionic order. (By assumption, $\mathrm{End}_{\overline{\mathbb{F}}_p}(E_i) = \mathrm{End}_{\mathbb{F}_{p^2}}(E_i)$ so the notation is also unambiguous). By the "supersingular case", we will always refer to this situation, i.e. when considering a supersingular elliptic curve $E/\mathbb{F}_{p^2}$ we will always assume it is a maximal or minimal curve.

In that case, ϕ is also always represented by the ideal $I = I(\phi)$, i.e. we also have $\mathrm{Ker}\,\phi = E[I]$: this is the well known Deuring correspondence (see [Ler22b] for a nice overview, and [Voi21] for a complete reference). Furthermore, $\mathrm{End}(E_1)$ is the left order associated to I and $\mathrm{End}(E_2)$ its right order.

In both the oriented and supersingular cases, the dual/contragredient isogeny $\widetilde{\phi_I} : E_2 \to E_1$ is represented by the ideal $\bar{I}$, and we let $N(I)$ be the integer such that $I\bar{I} = \bar{I}I = N(I)$, so that ϕ_I is a $N(I)$-isogeny. The number $N(I)$ is the norm of I in the quadratic case, and its reduced norm in the quaternionic case.

It is well known that supersingular isogeny graphs are expander (and even Ramanujan in dimension 1). This means that following a sufficiently long path of small degree isogenies, the codomain elliptic curve is close to statically uniform among all supersingular curves. Furthermore, since the endomorphism rings are not commutative, there is no commutative group action (we only have a groupoid of ideal classes), so Kuperberg's attack does not apply in the supersingular setting. However, this absence of commutativity makes the key exchange more difficult: given two isogenies $\phi_1 : E_0 \to E_1$ and $\phi_2 : E_0 \to E_2$ where the isogenies ϕ_i are secret and the domains and codomains curve are public, we could still define a common secret via a pushforward square $\phi : E_0 \to E_{12}$, but the curve E_{12} depends on the choice of ϕ_1, ϕ_2: using other isogenies $\phi'_1 : E_0 \to E_1, \phi'_2 : E_0 \to E_2$ with the same codomains

E_1, E_2 could give another result E'_{12}. We will go back to this when we look at Eichler orders in Appendix C.2.2. The SIDH protocol used the clever solution to go around this problem by publishing extra "torsion information" on ϕ_1, ϕ_2 to allow for Alice and Bob to compute the pushforward isogenies $E_1 \to E_{12}, E_2 \to E_{12}$. We refer to [De 17] for a nice introduction. Unfortunately (but fortunately for the HD representation!), this extra information turned out to reveal too much information.

4.1.3. *Orientations.* In [CK20], Colo and Kohel introduce the notion of orientations on the supersingular isogeny graph. With E_1, E_2 are supersingular curves over $\mathbb{F}_{p^2}$ as above, and given a quadratic imaginary order R, an R-orientation on E_i is an embedding $R \hookrightarrow \mathrm{End}(E_i)$. The isogeny ϕ is said to be R-oriented if it commutes with the orientations on E_1 and E_2. If R is saturated in $\mathrm{End}(E_1)$ (the orientation is also said to be primitive), we obtain an action by the class group of R on the oriented horizontal isogeny graph from E_1 (like we had for the horizontal isogeny graph of ordinary curves with $R = \mathrm{End}(E_1)$). This is convenient construction to build a group action on elliptic curves with a controlled number of points: $E_i(\mathbb{F}_{p^2}) = \mathbb{Z}/(p \pm 1)\mathbb{Z}^2$ (depending on the quadratic twist used). More generally, allowing going up R-oriented isogenies we obtain a volcano structure.

An important example is when E_1 is defined over $\mathbb{F}_p$, in that case $\mathrm{End}_{\mathbb{F}_p}(E_1)$ is only a quadratic imaginary order containing $\mathbb{Z}[\pi_p]$ (which is either maximal or of index 2 in its maximal order O_p), and we have a natural orientation given by the Frobenius endomorphism π_p. This is the orientation at play for the CSIDH cryptosystem, and we have a free transitive group action on $\mathrm{Cl}(\mathbb{Z}[\pi_p])$ or $\mathrm{Cl}(O_p)$ (depending on whether $\mathrm{End}_{\mathbb{F}_p}(E_1) = \mathbb{Z}[\pi]$ or O_p) on the supersingular curves $E/\mathbb{F}_p$ at the same 2-volcano level as E_1. This situation is thus very similar to the ordinary case, where the Frobenius also provides a natural orientation. More generally, looking at the graph of supersingular elliptic curves E d-isogeneous to their Galois conjugate: $\phi_d : E \to E^{(p)}$ (with isogenies respecting this morphism) is essentially equivalent to looking at an orientation by $\sqrt{-dp}$. The case $d = 1$ corresponds to the case supersingular curves defined over $\mathbb{F}_p$ [CS21].

More recently, other orientations on supersingular curves have been considered, for SCALLOP [FFK+23]. Here the orientation is given by a specific endomorphism $\alpha \in \mathrm{End}(E_1)$ rather than the Frobenius, so one needs to provide an efficient representation of α.

One can see orientations as a way of providing volcano like graph structure on top of supersingular isogeny graphs (an orientation is an extra structure in the terminology of categories). The notion of orientation allows to treat in a unified setting both ordinary curves and supersingular curves over $\mathbb{F}_p$ (which are both naturally oriented by the Frobenius), and even more general cases like SCALLOP.

A word of warning: for general R-oriented curves, if p is inert in R (this does not happen in the ordinary case or for supersingular curves over $\mathbb{F}_p$), then not all horizontal isogenies come from an ideal: the Frobenius isogeny π_p is horizontal and has degree p, and cannot come from an ideal of R since there are none of norm p. This is essentially the only obstruction [Onu21]. In particular, the class group action $\mathrm{Cl}(R)$ is not quite transitive in that case: there are two orbits.

Remark 4.1. The functor from ideal to isogenies extends to R-modules, where R is a primitive orientation on E. More precisely, one can build a (contravariant) functor from modules to group schemes and module maps to group morphisms, see [JKP+18] for ordinary and supersingular elliptic curves, and its extension to the oriented case in [PR23a]. This general module setting has several advantages. First, this functor, applied to a projective R-module

of rank g produces an abelian variety of dimension g isogeneous to E^g; so using modules allow to understand the higher dimensional isogeny graph of E^g. This was used for instance to find the formulas for Clapotis [PR23b].

Secondly, non projective modules encode level structure informations. This can be used to handle level structure at the module level, we will see an example of this in Appendix D.

Finally, for supersingular curves, it can be used to understand the "forgetting the orientation" functor; from the module point of view if O is the full endomorphism ring and R an orientation, forgetting the orientation amount to sending M to $M \otimes_R O$. This is, implicitly, the approach used in [ACL+23; ACL+22] to study the interaction between the R-oriented isogeny graph and the full supersingular graph.

4.2. **The supersingular case.** Recall that if E_1, E_2 are supersingular elliptic curves defined over $\mathbb{F}_{p^2}$, with all their geometric endomorphisms defined over $\mathbb{F}_{p^2}$ then any isogeny $\phi : E_1 \to E_2$ is represented by an ideal I.

We will assume that we have an efficient representation of the endomorphism ring $\mathrm{End}(E_1)$. This means that we both know its representation as an abstract maximal O_1 order in $B_{p,\infty}$, the quaternion algebra ramified at p and infinity, and that we are also able to evaluate endomorphisms $\alpha \in O_1$ on points of E_1. Typically this is done by giving an efficient representation of a basis $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ of the $\mathbb{Z}$-lattice O_1, basis given by endomorphisms of reasonable degree (polynomial in p). (In practice, we take $\alpha_1 = 1$.)

We remark that given an effective representation of a basis α_i, it is possible to recover the abstract structure of $\mathrm{End}(E_1)$ by computing Weil pairings. Conversely, given an abstract representation of O_1, it is possible to recover an effective representation for the endomorphisms: this is done by starting with an elliptic curve E_0 with an effective representation, computing a suitable effective isogeny between E_0 and E_1, and transporting the effective endomorphisms representation from E_0 to E_1. We refer to [EHLMP18; Wes22] for more details.

Given this effective representation of $\mathrm{End}(E_1)$, the ideal I provides a compact representation of the isogeny ϕ. Furthermore, $\ker \phi = E[I]$ so, by Section 3.2, this representation is effective if $n = N(I)$ (where N denote the reduced norm) is smooth and the N-torsion is accessible. See Example A.5 for more details on how to compute generators for $E[I]$.

In fact, the ideal representation is always effective, even when $N(I)$ is a large prime. Indeed, for isogeny based cryptography, many algorithms of the form `IdealToIsogeny` have been proposed, to make the Deuring correspondence between ideals and isogenies not only effective but practical.

We can broadly distinguish between four (minor) generations of these algorithms (we refer to Appendix C for more details):

(1) The KLPT [KLPT14] algorithm can be (heuristically) used to smoothen in polynomial time an arbitrary ideal I into a smooth equivalent ideal J of large norm. A proven version (under GRH) is given in [Wes22]. The first generation then takes J to be powersmooth (or smooth with accessible $N(J)$-torsion). We can then reconstruct the associated isogeny ϕ_J from its kernel $E[J]$. From ϕ_J, we can recover ϕ_I.
(2) The idea for the second generation is to take J to be of (reduced) norm 2^e rather than powersmooth, and to select a prime p such that a large 2^f-torsion is rational on supersingular elliptic curves $E/\mathbb{F}_{p^2}$. This gives the first version of the Deuring correspondence that is practical enough to be used in a signature protocol: SQIsign [DKLPW20].

A problem that has to be solved is that since $2^f \mid p \pm 1$, we have $2^f < p$ while the KLPT algorithm gives (generically) an ideal of norm $p^{4.5} \gg p$. The solution is to cut the isogeny represented by J into chunks of 2^f-isogenies, so it gets decomposed as $\phi_J : \phi_m \circ \ldots \circ \phi_1$. The problem is then to find the kernel of the intermediate isogenies $\phi_i : E_i \to E_{i+1}$, this is also known as "refreshing the 2^f-torsion".

The solution used by the second generation is to construct isogenies $E_0 \to E_i$ from a special elliptic curve E_0 to refresh the torsion; this is combined with an improved version of KLPT which use the theory of Eichler orders to decrease the degree produced by the KLPT smoothening algorithm from $\approx p^6$ to $\approx p^{4.5}$.

A further improvement to this method is described in [DLLW23], where a suitable endomorphism is constructed on each intermediate curve E_i to refresh the torsion, rather than via a special isogeny $E_0 \to E_I$

We refer to [Ler22b] for more details on this second generation.

(3) In the third generation, one still compute an equivalent ideal J of large reduced norm 2^e, which is then split into chunks of 2^f-isogenies, but the ideas of the HD representation are used there to refresh the torsion.

In [Ler23b], Leroux explain how to use a dimension 2 representation of arbitrary endomorphisms (similar to the techniques of SCALLOP-HD) to refresh the torsion on E_i, while in [ON24], Onuki and Nakagawa, inspired by SQIsignHD, use isogenies $E_0 \to E_i$ as in the original SQIsign version, but represented in dimension 2.

(4) In the fourth generation, the smoothening KLPT step is completely bypassed, and the ideal I is directly converted into an isogeny, using higher dimensional isogenies like for the third generation.

This algorithm (Clapotis, see Example 6.13) was introduced in [PR23b] to convert an ideal in the oriented case (see Section 4.3), and adapted to the supersingular setting for SQIsign2d in [BDD+24].

In the algorithms above, rather than computing $\phi_I : E_1 \to E_2$ directly, we often compute another equivalent but easier isogeny $\phi_J : E_1 \to E_2$. To recover ϕ_I from ϕ_J, one can use a double path algorithm, see Appendix C.1.

Conceiving improved ideal to isogeny algorithms that are not only effective in theory, but really practical, has thus been a subject of intense research that has made tremendous progress since the HD representation was introduced.

We have seen that in the fourth generation, used in the higher dimensional versions of SQIsign, like SQIsignHD [DLRW24] or the even more recent SQIsign2d variants [BDD+24; NO24; DF24], the KLPT "smoothening" algorithm is no longer needed. A hope for a fifth generation would be to find a way to improve KLPT to generate smooth ideals of norm $\approx p$ rather than $\approx p^{4.5}$, see Appendix C.5.

Remark 4.2. The main drawback of the ideal representation is that it requires the knowledge of $\mathrm{End}(E)$. Furthermore, giving the ideal I leaks the endomorphism ring (which is the left order of I), so one cannot share this ideal representation in settings where $\mathrm{End}(E)$ is supposed to remain secret, like in isogeny based cryptography. For that reason, Leroux introduced in [Ler22a] *the suborder* representation, which is an isogeny representation which is still efficient while only leaking part of the endomorphism ring. This representation has now been essentially superseded by the HD representation which leaks no informations on the endomorphism ring.

4.3. **The oriented case.** Let's assume that we have an efficient primitive orientation on E_1 by a quadratic imaginary order R. We recall that this includes the case of ordinary elliptic

curves and supersingular curves over $\mathbb{F}_p$, where in both cases R is given by the saturation of $\mathbb{Z}[\pi]$ in $\mathrm{End}_{\overline{\mathbb{F}}_q}(E)$.

Like in the supersingular case, converting an horizontal (i.e. invertible) ideal $I \subset R$ into an horizontal isogeny $\phi_I : E_1 \to E_2$, requires to compute the kernel $E[I]$ and then use the algorithms from Section 3.2. This is only effective if $N(I)$ is smooth and the $N(I)$-torsion is accessible.

In the supersingular case, we saw that it was always possible, thanks to the KLPT algorithm, to efficiently *smoothen* I, i.e., find in polynomial time an equivalent ideal J of smooth norm. In the ordinary case, this is much harder. We know that (under GRH), $\mathrm{Cl}(R)$ is generated by ideals of norm $O(\log^2 \Delta_R)$. So one can definitively decompose I into an equivalent smooth ideal J of smoothness bound $O(\log^2 \Delta_R)$, and heuristically this holds for a powersmooth decomposition too (increasing the smoothness bound if needed). But as we will explain below, we currently only have subexponential algorithms to smoothen an ideal in the oriented case.

If we can find J, then once we evaluate ϕ_J, we can evaluate ϕ_I if needed as we did in the supersingular cases, by using the effective R-orientation on E_1 (like in Appendix C.1, this may involve constructing a double path or using a division algorithm if we want to evaluate ϕ_I on points of torsion non coprime to $N(J)$).

If R is not given by the natural Frobenius orientation, we also need to push the R-action into an effective representation on E_2. There are two ways to do that: if we have an effective orientation from some special curve E_0, then we can build a double path from E_0 to E_1, E_2 in order to propagate the orientation to each. The problem is that this way of representing the orientation leaks an isogeny path from E_0 to E_i, which we want to avoid in cryptographic applications of isogenies. The other solution, if $R = \mathbb{Z} \oplus \mathbb{Z}\alpha$ is to simply give an HD representation of α on each E_i.

Like for the supersingular ideal to isogeny algorithms, we can distinguish four minor generations in the oriented case:

(1) The first generation simply bypass the smoothening problem by considering only ideals of the form $I = \prod \mathfrak{l}_i^{e_i}$ where the $\mathfrak{l}_i$ are prime ideals of small norm and the exponents e_i are small. In other words: we consider a *restricted group action* [ADMP20; DHK+23]. This is sufficient for a key exchange like CSIDH [CLMPR18], and even for signatures via rejection sampling [DG19].
(2) The second generation is to do a huge class group computation, using the standard classical subexponential algorithms. This is the approach of CSI-FiSh [BKV19].
(3) The third generation is to take an orientation by a non maximal conductor, in order to simplify the class group computation. A question is then how to represent the endomorphism α giving the orientation; in SCALLOP α is taken to be of smooth-norm, while in SCALLOP-HD α is arbitrary and represented via a dimension 2 HD isogeny.
(4) The fourth generation, Clapoti(s) [PR23b] completely bypass the smoothening step (this is the same method as the fourth generation in the supersingular case).

As we will explain below, even when the class group is easy to compute as in SCALLOP-HD, the smoothening step is still subexponential. By contrast, Clapoti(s) allows to compute the action of an invertible ideal I in polynomial time (in $\log p$ and $\log \Delta_R$) through higher dimensional isogenies. So once again, the HD representation proves crucial.

Like in the supersingular case, we now have a polynomial time algorithm to translate I into an isogeny, when I is *invertible*. In summary: the ideal representation is both compact

and efficient, both in the supersingular case, or in the oriented *horizontal* case. It is still unknown how to ascend or descend a large ℓ-isogeny volcano in polynomial time in $\log \ell$, see Section 7.

Remark 4.3. Let R be a primitive orientation on E_0, and $I \subset R$ an invertible ideal. Although we do not know how to smoothen in polynomial time an ideal $I \subset R$ in the oriented case, it is possible to smoothen in polynomial time the module map $I \oplus \bar{I} \to R \oplus R$. Technically, to handle the polarisations, we work with Hermitian modules and similitudes. Here the Hermitian form on R is the one induced by the norm,, the one on $R \oplus R$ is the product Hermitian form and the Hermitian form on $I \oplus \bar{I}$ comes from pullback. And for any N large enough (heuristically[9] $N \gg \min(N(I), \Delta_R^{1/2})^2 \Delta_R^2$), we can find a module map ϕ which is a N-similitude for these Hermitian form. Taking $N \gg p^3$ smooth gives our smoothening. Using the module equivalence of category Remark 4.1, we can convert this smoothened module map into an N-isogeny $E_0 \times E_0 \to E_I \times E_{\bar{I}}$ with respect to the product principal polarisations. More details will be given in [PR23a].

4.3.1. *Smoothening an ideal.* We give more details on the smoothening step used by the second and third generations. We first want to find $I \sim J = \prod \mathfrak{l}_i^{e_i}$ for small prime ideals $\mathfrak{l}_i$ of norm ℓ_i; this is already hard classically.

Furthermore, if the e_i are too large, so that the $\ell_i^{e_i}$ torsion is not accessible, we need to split the $\mathfrak{l}_i^{e_i}$ isogeny into chunks of $\mathfrak{l}_i^{f_i}$ isogenies, that we compute one by one. The problem is that if the exponent e_i is too large, we might need superpolynomially many chunks. We can reformulate this problem by saying that, even when $\mathfrak{l}$ is small, the action of $\mathfrak{l}^n$ takes time $O(n)$ to compute rather than $O(\log n)$. So we really want to find a powersmooth decomposition, i.e. such that the e_i are small. This is hard even with a quantum computer.

The current solution is as follows:

(1) Find the group structure of $\mathrm{Cl}(R)$ with respect to small generators $\mathfrak{l}_i$, and in particular the lattice of relation L.
(2) Find a decomposition $I \sim \prod \mathfrak{l}_i^{e_i}$, with potentially very large e_i. In practice, this step can be bypassed when I is directly given as $I = \prod \mathfrak{l}_i^{e_i}$ with large e_i (like for some signatures schemes).
(3) Solve (an approximation of) the close vector problem on the lattice of relations L, to obtain $I \sim \prod \mathfrak{l}_i^{e'_i}$ where the vector of exponents (e'_i) is small. To get a good approximation of CVP requires to find a good short basis of the lattice of relations L (the better the basis, the better the CVP solution).

4.3.2. *Class group computation.* For a generic quadratic order, the best classical algorithms to compute the class group of R are in subexponential time $L_{1/2}(\Delta_R)$.

The classical algorithm first finds relations between smooth ideals with a $L_{1/2}(\Delta_R)$ smoothness bound. Then a linear algebra step constructs the lattice of relations with respect to generators of norm $O(\log^C \Delta_R)$ for some appropriate constant C. We also have polynomial time quantum algorithms.

In practice, as shown by [BKV19] which gave a record class group computation for the CSI-Fish signature, this class group computation can be done for Δ_R of around 512 bits.

[9] The similitude with the KLPT algorithm from Appendix C.2 is not a coincidence, in both case the lattice of endomorphisms (of the special curve E_0 for KLPT and of R-oriented endomorphisms on $E_0 \times E_0$ in our situation) is of rank 4 and has elements of small norms.

To go further (classically), the idea of SCALLOP [FFK+23] is to use special quadratic orders R, with conductor f carefully chosen inside a maximal order R_0 of small discriminant Δ_{R_0}.

From the conductor square (see Appendix D), we obtain the exact sequence from Equation (4), which shows that computing $\mathrm{Cl}(R)$ amounts to computing the group structure of $\mathrm{Cl}(R_0)$ and of $(R_0/fR_0)^*/(R/fR)^*$. Since R_0 is chosen to have small discriminant, computing $\mathrm{Cl}(R_0)$ is easy. If $f = \ell$ is a prime, the second part amount to computing discrete logarithms in $\mathbb{F}_\ell^*$ if ℓ splits in R_0, and in $\mathbb{F}_\ell^{2,*}/\mathbb{F}_\ell$ if ℓ is inert in R_0. Carefully choosing f (typically f a large prime that splits in R_0 and such that $f-1$ is smooth so that DLPs in $\mathbb{F}_\ell$ are easy) allows to compute the large class group $\mathrm{Cl}(R)$ in practice.

In SCALLOP [FFK+23], due to the constraints of having to give an efficient representation of the orientation, f could be chosen smooth, but with a quite large smoothness bound. In SCALLOP-HD [CLP24], the authors show that switching to a dimension 2 HD representation for the orientation allows to take $f = \ell = 2^m u + 1$ with a small u, which allows for very fast DLPs in $\mathbb{F}_\ell^*$.

4.3.3. *Decomposing an ideal.* Decomposing an ideal I into an equivalent ideal $I \sim \prod \mathfrak{l}_i^{e_i}$ (where $\mathfrak{l}_i$ are our chosen small generators for $\mathrm{Cl}(R)$) is very similar to the class group computation, except it needs to be done online, this is not a precomputation. The classical algorithm is in $L_{1/2}(\Delta_R)$ for a generic quadratic order R, while quantum algorithms are polynomial time. If R is a special order of conductor f inside a maximal order R_0 of small discriminant, and such that $f = \ell$ is a large prime that splits in R_0 and such that $\ell - 1$ is smooth, like in SCALLOP or SCALLOP-HD, the decomposition amounts to a multi-DLP in $\mathbb{F}_\ell^*$, which is easy.

4.3.4. *Closest vector problem.* Since we have constructed the lattice of relations L between the $\mathfrak{l}_i$, we can use this lattice to shrink the exponents e_i: this is a closest vector problem. We can solve (an approximation of) the close vector problem on L by first finding a good short basis of L.

Heuristically, as explained in https://yx7.cc/blah/2023-04-14.html, using a lattice reduction algorithm of complexity $\approx L_\alpha(\Delta_R)$ allows to find exponents bounded by $\approx L_\beta(\Delta_R)$, where $\beta = \frac{1}{2} - \frac{1}{2}\alpha$ (and neglecting polynomial factors in $\log \Delta_R$), using $d \approx \log^{1-\beta} \Delta_R$ many small ideals (so working with a lattice of dimension d). We find out that, using a lattice reduction which takes polynomial time like LLL, we get exponents of size $L_{1/2}(\Delta_R)$. This is because we are in the special case of class groups of quadratic imaginary field, so we can expect the class group to be almost cyclic, and we heuristically expect it to be generated by very small elements of norm up to $\log^{1/2}(\Delta_R)$, so even through the LLL algorithm gives an exponential approximation factor in the dimension of the lattice, this dimension is small enough that the resulting approximation factor is subexponential in Δ_R. With a lattice reduction taking time $L_{1/3}(\Delta_R)$, we get exponents of size $L_{1/3}(\Delta_R)$. With a lattice reduction taking time $L_{1/2}(\Delta_R)$, we get exponents of size $L_{1/4}(\Delta_R)$, and with a lattice reduction taking exponential time we get exponents of size $O(1)$. And quantum algorithms do not seem to help for this step, because we do not know how to exploit that the lattices relations come from a class group.

This means that asymptotically, when using the generation 2 algorithm to compute the full class group of $\mathbb{Z}[\sqrt{-p}]$ like in CSI-FiSh, since this step takes $L_{1/2}(\Delta_R)$ already we can allow the precomputation phase to be of $L_{1/2}(\Delta_R)$ and expect exponents of size $L_{1/4}(\Delta_R)$. However, in a setting like SCALLOP or SCALLOP-HD where the class group computation is taylored to be easy (polynomial time), we still need a $L_{1/3}(\Delta_R)$ precomputation to get

exponents of size $L_{1/3}(\Delta_R)$ (or we could spend more precomputation time to get smaller exponents).

In practice, for the specific examples of class group action as computed in CSI-FiSh, SCALLOP, or SCALLOP-HD, the lattice reduction phase was not the bottleneck (for CSI-Fish it was computing the lattice itself, step bypassed in SCALLOP or SCALLOP-HD, at the cost of no longer having a natural orientation). But this asymptotic bottleneck prevents an asymptotic instantiation of the class group action in polynomial time, even with a quantum computer.

Only the fourth generation, which bypass the smoothening step, can give a fully unrestricted group action.

5. The HD representation

The ideal representation is our first isogeny representation that gives an effective representation for horizontal isogenies, provided that we know an effective representation of the endomorphism ring or orientation of the domain.

We have seen in Section 3.3 that the interpolation representation can give a compact representation of *any* isogeny. However, recovering the isogeny from the interpolation data is at least in $\widetilde{O}(n)$.

Recently, as a side product of the SIDH attacks [CD23; MMPPW23; Rob23b], a much more sophisticated algorithm has been found to reconstruct the isogeny ϕ from suitable interpolation data, via higher dimensional isogenies.

The most flexible version is the following [CDM+24]:

Theorem 5.1. *Let $\phi : E_1 \to E_2$ be an n-isogeny (with as usual n prime to p). Let G be a subgroup of E_1 of order at least $4n+1$, and $(P_1, \dots, P_r)$ generators of G. Given a point $P \in E_1(\mathbb{F}_{q^r})$ and the interpolation data $(P_i, f(P_i))$, there is an algorithm to evaluate $\phi(P)$ in time polynomial in $\log n$, r, $\log q$, the largest prime factor of n, the extension degrees of the field of definition of P and the points P_i, and the extension degree of the points of $\ell^{\lfloor e/2 \rfloor}$ torsion for each prime power $\ell^e \mid \#G$.*

For our applications, we will only need a simplified version where we assume that we have interpolation data on the full N-torsion, version contained in [Rob23b] and which was exploited in [Rob22a] for isogeny representations. In that setting, Theorem 5.1 can be rephrased as follows: assume that we have interpolation data on the N-torsion with N large enough ($N > n$, but see Appendix B for a relaxation to $N^2 > n$), smooth and the N-torsion accessible on E_1. Then we have an efficient representation of the isogeny ϕ. This is done by embedding ϕ into a N-isogeny Φ in higher dimension; we call this the *HD representation*.

Since the HD representation use isogenies of higher dimension, we first review isogenies between abelian varieties and algorithms to compute them in Section 5.1. We then explain Kani's lemma in Section 5.2, and how it can be used to embed isogenies to obtain the HD representation in Section 5.3.

Remark 5.2 (Level structure)**.** Theorem 5.1 shows that, in the context of isogeny based cryptography, revealing too much torsion information is insecure. Namely, a secret isogeny $\phi : E_1 \to E_2$ can be reconstructed efficiently if the adversary know its degree n, and its action on the N-torsion (provided that N is smooth and the N-torsion is accessible), as long as $N^2 \approx n$ (or, by Theorem 5.1, even just the action on a subgroup $G \subset E_1[N]$ of size $\approx n$). Still, revealing some torsion information (i.e., suitable image points under ϕ), as was done in SIDH, can be very useful to build isogeny based cryptosystems.

We discuss several solutions:

- Mask the torsion revealed. For instance in M-SIDH [FMP23] the torsion information $(\phi(P_1), \phi(P_2))$ is masked by some common scalar λ (one needs to be careful that the Weil pairing $e_{W,N}$ will reveal λ^2 modulo N). In FESTA [BMP23] the authors hide the torsion by a diagonal matrix. The correct notion under which to study these maskings is the concept of level structure, we refer to [DFP24] for an overview. An alternative, as in [Bas24] is to also hide the degree.
- Only reveal torsion information for a subgroup G of size much smaller than the degree n. This is for instance the approach for IS-CUBE and LIT-SIGAMAL [Mor23; Mor24].

  We recall that supersingular isogeny graphs are expander. This property remains true even when adding level structure information [BCC+23; CL23; PW24], at the condition of increasing the degree of the path relative to the degree of the revealed level structure. This means that we can heuristically expect that it is not easier to recover a degree n'-isogeny for which the action on the N-torsion is revealed than to recover a degree n-isogeny for which no torsion information is revealed, provided that $n' \approx nN^2$.
- Reveal the torsion information on the N-torsion for N large with respect to n, but prime rather than smooth. Indeed, in that case, while the embedding lemma still allows to embed ϕ into a higher dimensional N-isogeny Φ, without the smoothness condition we do not know how to evaluate Φ efficiently. This is an open question in dimension 1 already, see also Section 7. Still, this is quite a strong cryptographic assumption to make (that we don't know how to evaluate an isogeny from kernel generators of non smooth order), and we would recommend to use the other solutions if possible.

5.1. **Isogenies between abelian varieties.** The key idea behind Theorem 5.1 is to embed the n-isogeny ϕ into a higher dimensional N-isogeny Φ given by the interpolation data. We first briefly describe the type of isogenies we will work with, and existing algorithms to compute them.

If $\phi : A \to B$ is an isogeny, we denote by $\hat{\phi} : \hat{B} \to \hat{A}$ its dual isogeny.

Definition 5.3. An isogeny $\phi : (A, \lambda_A) \to (B, \lambda_B)$ between polarised abelian varieties is called an n-isogeny if the polarisation $\phi^*\lambda_B := \hat{\phi} \circ \lambda_b \circ \phi = n\lambda_A$.

We recall the following standard results, see [Rob23b].

Lemma 5.4. *Let $\phi : (A, \lambda_A) \to (B, \lambda_B)$ be an isogeny between principally polarised abelian varieties, and let $\widetilde{\phi} = \lambda_A^{-1} \circ \hat{\phi} \circ \lambda_B : B \to A$. Then ϕ is an n-isogeny if and only if $\widetilde{\phi} \circ \phi = n$ (or equivalently, $\phi \circ \widetilde{\phi} = n$).*

Furthermore, if $\Phi = \begin{pmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{pmatrix} : (A_1 \times A_2, \lambda_{A_1} \times \lambda_{A_2}) \to (B_1 \times B_2, \lambda_{A_2} \times \lambda_{B_2})$ is an isogeny between the product abelian varieties endowed with their product principal polarisations, then $\widetilde{\Phi} = \begin{pmatrix} \widetilde{\phi_{11}} & \widetilde{\phi_{21}} \\ \widetilde{\phi_{12}} & \widetilde{\phi_{22}} \end{pmatrix}$.

Remark 5.5. If $\phi : (A, \lambda_A) \to (B, \lambda_B)$ is an n-isogeny between principally polarised abelian varieties, the kernel $K = \operatorname{Ker} \phi \subset A[n]$ is maximal isotropic for the (polarised) Weil pairing $e_{W,n\lambda_A}$. And if n is prime (and $n \neq p$), $K(\bar{k}) \simeq (\mathbb{Z}/n\mathbb{Z})^g$. Conversely, if $K \subset A[n]$ is maximal isotropic for $e_{W,n\lambda_A}$, then the polarisation $n\lambda_A$ descends to a principal polarisation λ_B on $B = A/K$. Note however, that unlike the dimension one case, there can be several non equivalent principal polarisations on an abelian variety, so being an n-isogeny (with respect

to specific principal polarisations) is a stronger condition than just the kernel being maximal isotropic for $e_{W,n\lambda_A}$.

We have partial generalisations of the isogeny representations from Section 3 for elliptic curves to abelian varieties.

In low dimension, $g \leq 3$, any principally polarised abelian variety (over an algebraically closed field) is a Jacobian of a curve or a product of Jacobians. For instance in dimension 2 we have either a product of two elliptic curves or a Jacobian of an hyperelliptic curve of genus 2. In dimension 3, if (A,λ_A) is a principally polarised abelian variety with an indecomposable polarisation λ_A, then A is a Jacobian $A = \mathrm{Jac}(C)$, with C either hyperelliptic of genus 3 or a quartic curve.

The polarisation is important here: for instance every superspecial abelian variety $A/\mathbb{F}_q$ of dimension $g > 1$ is isomorphic (over the algebraic closure) as an unpolarised abelian variety to E_0^g, where $E_0/\overline{\mathbb{F}}_q$ is any supersingular curve, but in general a principal polarisation on A won't be given by the product principal polarisation. In dimension 2 over $\mathbb{F}_{p^2}$, among superspecial principally polarised abelian surfaces, there are roughly $\approx p^3/2880$ superspecial Jacobians, and $\approx p^2/288$ product of supersingular elliptic curves.

Starting in dimension 4, a generic abelian variety won't be a Jacobian. We can instead use the theory of algebraic theta functions, as developed by Mumford in [Mum66; Mum67a; Mum67b], with useful extensions by Kempf [Kem88; Kem89a; Kem89b; Kem90; Kem92] (see also [Mum83; Mum84; Mum91] for the analytic theory of theta functions).

5.1.1. *Vélu's like formulas.* We have an extension of Vélu's formula which allows to evaluate an n-isogeny $\phi : (A,\lambda_A) \to (B,\lambda_B)$ between principally polarised abelian varieties, given a generator representation $K = \langle T_1,\dots,T_g\rangle$ of its kernel, which is isotropic for the Weil pairing $e_{W,n\lambda_A}$, in the algebraic theta model induced by a symmetric theta structure of level m on A (this algebraic theta model is completely determined by the theta null point of A if $m \geq 4$; and if $m = 2$ we obtain a model for the Kummer variety $A/\pm 1$ if the polarisation is indecomposable). We note that the rank condition on the kernel K is similar to having a cyclic isogeny in dimension 1. Namely, by [LR12; CR15; LR22], in the case that n is prime to the level m and the characteristic p is coprime[10] to mn, [LR22] gives an algorithm in $O((mn)^g)$ arithmetic operations over the field of definition of the generators T_i to compute the theta null point of B, and also to compute the image of a point P (this time over the field of definition of P and the T_i). A sketch of an adaptation of this algorithm to the case when m is not coprime to n (this includes the important case where $n = 2$ is even) is given in [Rob21]. The formulas of [CR15] have been implemented in Magma [BCR10]; in the repository there is also a branch which contains the fork ThetaAV written by Anna Somoza and David Lubicz in Sage. (The faster formulas from [LR22] are currently only in the private development version.) These isogeny formulas are part of a research effort [LR12; CR15; LR15b; LR22; LR16; LR10; LR15a] to adapt the usual algorithms on elliptic curves (arithmetic, pairings, isogenies) to arbitrary abelian varieties represented by algebraic theta functions; see [Rob21] for an overview.

Remark 5.6 (Level 2 vs level 4)**.** In [Mum67a, § 6], Mumford constructs the universal abelian scheme of level m over $\mathbb{Z}[1/m]$ via Riemann's relations, when the level m is divisible by 4.

[10] A symmetric theta structure has even level, so this condition imposes that $p \neq 2$. But we can use the same lift and reduce technique as in [GL09] to handle the case $p = 2$. For instance [BCR10] contains code to compute dimension 2 isogenies in characteristic two, derived by adapting the formulas for characteristic 0 in such a way that they have good reduction modulo 2.

But level m requires m^g theta coordinates, so for efficiency we prefer to work in level $m = 2$. This adds some technical difficulties, notably for gluing and splitting, see [DMPR24, § 4.1].

For ℓ-isogenies between Jacobians, we also have an algorithm in $O(\ell^g)$ by [CE14], formulas which were optimised for $g = 2, 3$ in [Mil20] (see also [Tia24] for gluing and splitting isogeny formulas).

For small ℓ and in the Jacobian model in dimension 2, formulas for 2-isogenies were already known by Richelot [Ric36; Ric37], and specific formulas for the case $\ell = 3, 5$ were given in [BFT14; Fly15]. A general method to find ℓ-isogeny formulas in the generic Kummer model (rather than the theta model; this model was described by Cassel and Flynn [CF+96; Fly93]), are given in [Nic18] [CCS24]. Smith also gave 2-isogeny formulas in dimension 3 in [Smi08] when the domain is a Jacobian of an hyperelliptic curve (this allows to transfer the DLP problem to the Jacobian of a non hyperelliptic curve when the codomain is not a Jacobian of an hyperelliptic curve), see also [FK11]. Smith's approach to transfer the DLP from a Jacobian of a genus 3 hyperelliptic curve to a genus 3 quartic was generalized using the ℓ-isogeny formulas of [CE14; Mil20] in [Tia20].

5.1.2. *Decomposing a smooth isogeny.* We can then apply the same quasi-optimal strategies to decompose a smooth n-isogeny into a product of small isogenies as in dimension 1, when the n-torsion is accessible.

We give the statement for the theta model, since it handles all abelian varieties, but the same strategy works for isogenies between Jacobian models.

Lemma 5.7. *Let $(A, \lambda_A)/\mathbb{F}_q$ be a principally polarised abelian variety represented by a symmetric theta structure of level m. Let $K \subset A[n]$ be a maximal isotropic subgroup of rank g. Let $n = \prod_{i=1}^{a} \ell_i^{e_i}$, and $T_{1,i}, \dots, T_{g,i}$ generators of $K[\ell_i^{e_i}]$, which live in $\mathbb{F}_{q^{d_i}}$.*

If n is not coprime to m, we let $n' = n \vee m$ if n is odd and $n' = 2(n \vee m)$ if n is even, and we also need to assume that we are given points $T'_1, \dots, T'_g$ of n'-torsion above the generators of K, that are compatible with our theta structure. Then the T'_i induce a uniquely determined theta structure of level m on $B = A/K$, and we can compute the theta null point of B in time $\widetilde{O}(\sum_i e_i m^g \ell_i^g (d_i \log e_i + \sum_{j>i} d_i \vee d_j) \log q) = \widetilde{O}(a^2 d' e m^g \ell^g \log q)$, where e is a bound on the e_i, d' is a bound on the $d_i \vee d_j$, and ℓ is a bound on the ℓ_i.

Given a point $P \in A(\mathbb{F}_{q^{d_P}})$, represented by its theta coordinates, we can compute the theta coordinates of $\Phi(P)$ in $\widetilde{O}(\sum_i e_i m^g \ell_i^g (d_i \vee d_P) \log q) = \widetilde{O}(a e m^g \ell^g d'' \log q)$ where d'' is a bound on the $d_i \vee d_P$.

Proof. This follows from the general isogeny formulas from [Rob21], and the decomposition process is explained in more details in [DLRW24, Appendix F]. See Remark A.2 for our assumptions about the lattice of field extensions used to represent the CRT basis. □

Remark 5.8. The condition on having explicit points of n'-torsion in Lemma 5.7 is to ensure that the theta structure of $B = A/K$ is uniquely determined. We can relax this condition at the cost of taking some roots. For instance in level $m = 2$, to compute 2^e-isogenies in dimension 2, in [DMPR24; Rob23a] we describe an algorithm to compute them from only their kernel, by using appropriate square roots at the last two steps where we don't have enough information to determine the theta structure uniquely. These square roots are not needed if we have points of 2^{e+2}-torsion above the kernel.

5.1.3. *Fast formulas.* Due to the importance of the HD representation for dimension 1 isogenies, the specific case of 2^e-isogenies (or ℓ^e-isogenies with small ℓ) in dimension 2 and 4 has recently been the focus of optimisations. In dimension 2, [Kun22] Kunzweiler gives

efficient 2^e-isogeny formulas in the Jacobian model, which she has extended to the Kummer Jacobian model (private communication). In [DK23], Decru and Kunzweiler give efficient 3^e-isogeny formulas in the Kummer model. Recently, in [CCS24], Corte-Real Santos, Costello, Smith gave optimised 3-isogeny formula in dimension 2 in the theta squared Kummer model.

Fast 2^e-isogeny formulas in dimension 2 in the level 2 theta model (hence with theta Kummer surfaces) were given in [DMPR24]. These formulas achieve a factor 10 speed up for the codomain computation compared to Richelot isogenies, and a factor 30 for the image of points. These fast formulas are the basis of all the recent isogeny based cryptosystems which use a 2^e-HD representation in dimension 2; they are notably used for the verification of the SQIsign2d variants [BDD+24; NO24; DF24], which currently give the record verification time for the SQIsign family of signature schemes.

Fast formulas for 2^e-isogenies in the level 2 theta model in any dimension were sketched in the notes [Rob23a]; these have been fully worked out by Dartois in dimension 4 in [Dar24], along with explicit base change formulas. The main application of Dartois' work is for the verification of the SQIsignHD signature [DLRW24], which is done in dimension 4.

The reason we prefer to use as small a dimension as possible for the HD representation is that the level 2 theta model in dimension g requires 2^g coordinates, so we expect an exponential slow down with respect to the dimension. As a rule of thumb: in dimension g we work with 2^g theta coordinates, and to decompose a 2^e-isogeny we need to push g points in the kernel at each step, so we expect a time of $Cg2^g$ for the decomposition by step. The formulas are very similar (a combination of Hadamard transforms, squarings, and multiplication by suitable constants) across all dimensions, so we expect C to be roughly independent of g. So as a very rough approximation, we can expect dimension 2 2^e-isogenies to be 4 times slower than in dimension 1, dimension 4 isogenies to be 8 times slower than dimension 2 and 32 times slower than dimension 1, and dimension 8 isogenies to be 32 times slower than dimension 4, 256 times slower than dimension 2 and 1024 times slower than dimension 1.

In practice, a low level Rust and C implementations of the formulas of [DMPR24] need around $2 - 3ms$ to compute a 2^{128}-isogeny in dimension 2 over a field of 256 bits (the C implementation was written for SQIsign2d-West and will soon be available, the Rust implementation is already available). Compared in dimension 1, this is roughly a 4.5 slow down, this is expected with respect to our factor 4 slow down we argued previously, because in dimension 1 the best 2^e-isogeny formulas use Montgomery coordinates which are slightly faster than theta coordinates, and decompose the 2^e-isogeny into chunks of 4-isogenies, which is also slightly faster than decomposing into chunks of 2-isogenies.

Remark 5.9 (Fast Kummer surfaces versus generic Kummer surfaces)**.** As mentioned, the fast 2^n and 3^n isogeny formulas on Kummer surfaces from [DMPR24; CCS24] use Mumford's algebraic theta model (or twisted variants of this model, like the squared theta model of a Kummer surface). It is quite amazing that this theta model is so fast, since it is also universal and give models for abelian varieties in any dimension: Dartois Sage's implementation [Dar24] for 2^n-isogenies in dimension 4 is already very promising. One reason is that the algebraic theta model is taylored to have very efficient formulas for the map $(P, Q) \to (P + Q, P - Q)$; this is the duplication formula.

This efficient duplication formula (in level 2, hence on the Kummer variety) gives fast arithmetic: doubling and differential additions. This fast arithmetic has been exploited since a long time, notably for classical DLP-based cryptography in dimension 1 and 2 [CC86; Gau07; RSSB16; HR19]. This theta model (or its twisted variant given by squared thetas), is often referred as the fast Kummer model in the literature. It is thus not surprising

that the theta model is also used for fast isogenies in higher dimension. In fact, it was used even before the HD representation! In supersingular isogeny based cryptography, an isogeny $\phi : E_1 \to E_2$ over $\mathbb{F}_{p^2}$ induces via the Weil restriction of scalar functor an isogeny $W_{\mathbb{F}_{p^2}/\mathbb{F}_p}\phi : A_1 := W_{\mathbb{F}_{p^2}/\mathbb{F}_p}E_1 \to A_2 := W_{\mathbb{F}_{p^2}/\mathbb{F}_p}E_2$ defined over $\mathbb{F}_p$, where A_1, A_2 are principally polarised abelian surfaces (which are neither Jacobians nor product of elliptic curves over $\mathbb{F}_p$!). Furthermore, A_1, A_2 are 2-isogeneous over $\mathbb{F}_p$ to Jacobians $\mathrm{Jac}(C_1), \mathrm{Jac}(C_2)$ of hyperelliptic curves. In [Cos18], Costello explains that, using the squared theta model, computing the induced isogeny $\mathrm{Jac}(C_1) \to \mathrm{Jac}(C_2)$ in dimension 2 over $\mathbb{F}_p$ is potentially faster than computing the original isogeny ϕ in dimension 1 but over $\mathbb{F}_{p^2}$. This idea was recently revisited in the context of SQIsign in [CR24].

The main drawback of the theta model (of level 2) is that the fast duplication formula is obtained by splitting the map $(P, Q) \to (P + Q, P - Q)$ into two. This requires a specific Galois structure on the 2-Tate module, more precisely an abelian variety $A/\mathbb{F}_q$ has a rational theta model of level 2 if and only if there is a symplectic basis of $A[2]$ for the Weil pairing with trivial self Tate pairings [Rob21, § 2.11]. For general abelian varieties, the theta model is thus only defined over a field extension. In that case, at least for abelian surfaces, it can be useful to work with the Jacobian Kummer model or the generic Kummer model of Cassels and Flynn [CF+96].

5.1.4. *Other isogeny representations for abelian varieties.* For the other "classical" isogeny representations, the main difficulty is that while in dimension 1 we can (almost always) reduce to question about univariate polynomials or rational functions, in higher dimensions we need to work with multivariate rational functions, for which it is harder to obtain quasi-linear algorithms.

For instance, in theory, one could find a function representation of an isogeny ϕ in higher dimension by using any evaluation algorithm on the generic point η_A of A, but in practice equations for A are quite involved in higher dimension, so the function representation is not really used. Still, in dimension 2, when we have an isogeny $\phi : \mathrm{Jac}(C_1) \to \mathrm{Jac}(C_2)$ between Jacobian of hyperelliptic curves, it is convenient to express ϕ in terms of the Weierstrass coordinates of C_1, C_2. One method to do that is to evaluate ϕ on a formal point at precision 2, deduce a differential equation for ϕ, and to solve the differential system [CE14, § 6.2; KPR24; CMSV19], so at least this case is tractable.

We have the same problem with kernel equations: unlike the dimension 1 case where the kernel is described by a univariate polynomial, kernel equations in higher dimension are given by multivariate polynomials, which make them harder to work with. An algorithm to compute ϕ when we have a univariate parametrisation of the kernel K is given in [LR15b].

And of course, for the interpolation representation, we need multivariate rational function reconstruction. Likewise for adapting the `sqrtVelu` algorithm in higher dimension: the obstacles were informally worked out in a Ciao workshop meeting in Bordeaux in December 2022. The `sqrtVelu` algorithm is described by a resultant in [BDLS20], but can be reformulated as evaluating a degree $O(\sqrt{n})$ univariate polynomial P on $O(\sqrt{n})$ points, which can be done in $\widetilde{O}(\sqrt{n})$ by fast multipoint evaluation algorithms. We can extend this approach to higher dimension using the known isogeny formulas, but then we need fast *multivariate* multipoint evaluations, which exist in theory over finite fields by [KU11], but without practical implementations.

Finally, we also mention that reconstructing an isogeny from a root of the modular polynomial Φ_ℓ has been worked out in dimension 2 in [KPR24] (see Example E.1).

5.2. **Kani's lemma and its applications.**

Definition 5.10. A (n_1, n_2)-isogeny diamond is a commutative diagram of isogenies between polarised abelian varieties:

$$\begin{array}{ccc} A_0 & \xrightarrow{\phi_1} & A_1 \\ \downarrow{\scriptstyle \phi_2} & & \downarrow{\scriptstyle \phi_2'} \\ A_2 & \xrightarrow{\phi_1'} & A_{12} \end{array}$$

where $\phi_1 : A_0 \to A_1$ and $\phi_1' : A_2 \to A_{12}$ are n_1-isogenies, $\phi_2 : A_0 \to A_2$ and $\phi_2' : A_1 \to A_{12}$ are n_2-isogenies.

Remark 5.11. If n_1 is coprime to n_2, then an isogeny diamond as above is the same thing as a pushforward square from ϕ_1, ϕ_2 or a pullback square from ϕ_1', ϕ_2'.

We can now state Kani's lemma, which is contained in [Kan97, § 2, Proof of Th. 2.3].

Theorem 5.12 (Kani's lemma)**.** *Let n_1 and n_2 be two integers. Given a (n_1, n_2)-isogeny diamond, the isogeny $\Phi : A_0 \times A_{12} \to A_1 \times A_2$ given matricially by*

$$\Phi = \begin{pmatrix} \phi_1 & \widetilde{\phi_2'} \\ -\phi_2 & \widetilde{\phi_1'} \end{pmatrix}$$

is a (n_1+n_2)-isogeny between these product of abelian varieties with their product polarisations. If n_1 is coprime to n_2, the kernel of Φ is given by

$$\begin{aligned} \operatorname{Ker}\Phi &= \{(\widetilde{\phi_1}(P), \phi_2'(P)) \mid P \in A_1[n_1+n_2]\} \\ &= \{(-\widetilde{\phi_2}(P), \phi_1'(P)) \mid P \in A_2[n_1+n_2]\} \\ &= \{(n_1 P, \phi_2' \circ \phi_1(P)) \mid P \in A_0[n_1+n_2]\}. \end{aligned}$$

Proof. We compute $\widetilde{\Phi} \circ \Phi = \begin{pmatrix} \widetilde{\phi_1} & -\widetilde{\phi_2} \\ \phi_2' & \phi_1' \end{pmatrix} \begin{pmatrix} \phi_1 & \widetilde{\phi_2'} \\ -\phi_2 & \widetilde{\phi_1'} \end{pmatrix} = \begin{pmatrix} n_1+n_2 & 0 \\ 0 & n_1+n_2 \end{pmatrix}$ which shows that Φ is a (n_1+n_2)-isogeny for the product polarisations.

The kernel of Φ, of cardinal $(n_1+n_2)^{2g}$, is given by the image of $\widetilde{\Phi}$ on $(A_1 \times A_2)[n_1+n_2]$. If n_1 is coprime to n_2, the restriction of $\widetilde{\Phi}$ to $A_1[n_1] \times 0_{A_2}$ is injective so its image already spans the full kernel: $\operatorname{Ker}\Phi = \{(\widetilde{\phi_1}(P), \phi_2'(P)) \mid P \in A_1[n_1+n_2]\}$. The second equality follows by symmetry, and the third by plugging $P = \phi_1(P_0)$ with $P_0 \in A_0[n_1+n_2]$.

We refer to [MMPPW23, Theorem 1] or [Rob23b] for more details. □

Corollary 5.13 (Embedding an isogeny in dimension 2)**.** *Let $\phi : E_1 \to E_2$ be an n-isogeny. Let $N > n$ be coprime to n, $n' = N - n$, and assume that we know an efficient representation of an n'-isogeny $\psi : E_1 \to E_1'$. Then we can embed ϕ into an N-isogeny $\Phi = \begin{pmatrix} \phi & \widetilde{\psi'} \\ -\psi & \widetilde{\phi'} \end{pmatrix} : E_1 \times E_2' \to E_2 \times E_1'$. Furthermore, if N is smooth, the N-torsion is accessible on E_1, and we know how ϕ acts on $E_1[N]$, then Φ can be efficiently computed.*

We have $\phi = p \circ \Phi \circ i : E_1 \to E_1 \times E_2' \to E_2 \times E_1' \to E_2$ where $i(P) = (P, 0)$ and $p(P, Q) = P$, so we can recover $\phi(P)$ via $\Phi((P, 0)) = (\phi(P), -\psi(P))$.

Similar constructions hold when we have $\widetilde{\psi} : E_1' \to E_1$, $\psi' : E_2 \to E_2'$ or $\widetilde{\psi'} : E_2' \to E_2$.

Proof. Given $\psi : E_1 \to E_1'$ of degree n' coprime to n, we can look at the pushforward square, which form a (n, n')-isogeny diamond:

$$\begin{array}{ccc} E_1 & \xrightarrow{\phi} & E_2 \\ \downarrow{\scriptstyle \psi} & & \downarrow{\scriptstyle \psi'} \\ E_1' & \xrightarrow{\phi'} & E_2' \end{array}$$

We then have $\Phi : E_1 \times E_2' \to E_2 \times E_1'$ a N-isogeny. Furthermore, the kernel of $\widetilde{\Phi}$ is given by $\operatorname{Ker} \widetilde{\Phi} = \{(\phi(P), -\psi(P)) \mid P \in E_1[N]]\}$. From our assumption on ϕ, and since ψ is supposed to be efficient, we can compute this kernel, then compute $\widetilde{\Phi}$, and then compute Φ whose kernel is $\widetilde{\Phi}((E_1' \times E_2')[N])$.

If we now assume that we know an efficient representation of $\widetilde{\psi} : E_1' \to E_1$, then we can recover how ψ acts on $E_1[N]$, which is enough to get $\operatorname{Ker} \widetilde{\Phi}$.

If we are given $\psi' : E_2 \to E_2'$ (or simply its action on the N-torsion), we can directly recover $\operatorname{Ker} \Phi = \{(\widetilde{\phi}(P), \psi'(P)) \mid P \in E_2[N]]\} = \{(n_1 P, \psi' \circ \phi(P)) \mid P \in E_1[N]\}$. (The first equality requires to extract the action of $\widetilde{\phi}$ on $E_2[N]$ from the action of ϕ on $E_1[N]$, but not the second.) Finally if we are given $\widetilde{\psi'}$, we can just extract the action of ψ' on $E_2[N]$. □

Example 5.14. Let $p \equiv 3 \mod 4$ and $E_0 : y^2 = x^3 + x / \mathbb{F}_p$, this is a supersingular elliptic curve whose endomorphism ring over $\mathbb{F}_{p^2}$ contains $\mathbb{Z}[i, \pi_p]$ with $i^2 = -1$ and $\pi_p^2 = -p$. In particular, we can build endomorphisms $x + yi + u\pi_p + vi\pi_p$ of norm $x^2 + y^2 + p(u^2 + v^2)$. So on this special curve, if n' is a sum of two squares or n' is large enough ($n' \gg p$), we can build dimension 1 endomorphisms of norm n'. More generally, a similar strategy works for a supersingular curve which contains a quadratic order of small discriminant. We will see in Example 6.12 how to leverage this construction of endomorphisms of large enough degree to construct isogenies from E_0 of any degree.

It remains to find an efficient representation of some isogeny $\psi : E_1 \to E_1'$ of some fixed degree. The main insight is that this is always possible, provided we go in higher dimension (this is Zahrin's trick).

Proposition 5.15. *Let n' be an integer. If $n' = a_1^2$, then $[m] : E_1 \to E_1$ is an n'-isogeny.*

If $n' = a_1^2 + a_2^2$, then $\begin{pmatrix} a_1 & a_2 \\ -a_2 & a_1 \end{pmatrix} : E_1^2 \to E_1^2$ is an n'-isogeny.

If $n' = a_1^2 + a_2^2 + a_3^2 + a_4^2$ (which is always the case), then $\begin{pmatrix} a_1 & -a_2 & -a_3 & -a_4 \\ a_2 & a_1 & a_4 & -a_3 \\ a_3 & -a_4 & a_1 & a_2 \\ a_4 & a_3 & -a_2 & a_1 \end{pmatrix} :$ $E_1^4 \to E_1^4$ is an n'-isogeny.

Proof. If we let M be the matrix of endomorphism appearing in the proposition, then the contragredient isogeny $\widetilde{M}$ for the product polarisation is given by $\widetilde{M} = \overline{M}^T$ by Lemma 5.4, and we check that $\widetilde{M}M = n' \operatorname{Id}$. □

Example 5.16 (Embedding an isogeny in higher dimension)**.** Let α be the matrix from Proposition 5.15, which induces an n'-endomorphism on E_1^u and E_2^u, where $u = 1, 2, 4$ according to whether n' is a sum of $1, 2, 4$ squares. Then as in Corollary 5.13, we can embed ϕ into the dimension $g = 2u$ endomorphism $\Phi = \begin{pmatrix} \phi \operatorname{id}_u & \tilde{\alpha} \\ -\alpha & \widetilde{\phi} \operatorname{id}_u \end{pmatrix} : E_1^u \times E_2^u \to E_2^u \times E_1^u$.

Then Φ embeds ϕ (and its dual): $\phi = p \circ \Phi \circ i : E_1 \to E_1^u \times E_2^u \to E_1^u \times E_2^u \to E_2$, and has kernel $\operatorname{Ker}\Phi = \{(-\tilde{\alpha}(P), \phi\,\mathrm{id}_u(P)) \mid P \in E_1^u[N]\} = \{(nP, \alpha\varphi\,\mathrm{id}_u(P)) \mid P \in E_1^u[N]\}$.

Furthermore, in that case we can split Φ in two, see Appendix B, which is not possible in Corollary 5.13 (unless we already know all four curves E_1, E_2, E_1', E_2').

Remark 5.17 (Choice of dimension)**.** Due to the cost of computing isogenies in high dimension, for efficiency we would like to use Example 5.16 with g as small as possible. In practice, there are sufficiently many integers which are sum of two squares that it is very rare that we need to go all the way to $g = 8$, usually $g = 4$ is sufficient.

Furthermore, in the case where E_1 (or E_2) is the special curve from Example 5.14, then we can use endomorphisms γ of the form $a_1 + a_2 i$ on E_1 to relax the dimension u in Proposition 5.15: we can use $u = 1$ when n' is a sum of two squares and $u = 2$ in the general case when n' is a sum of four squares. This means that in that case we have $g = 2$ or 4 rather than 4 or 8. However, the pushforward of γ by ϕ will in general be an isogeny rather than an endomorphism, so this means that we cannot split Φ in two in that case if we use endomorphisms like γ rather than integers in our matrix α from Example 5.16. A similar strategy works for a curve containing a quadratic order with small discriminant.

Remark 5.18. It is not always easy to check if an integer n' is a sum of two squares (and if so find the decomposition). This usually requires to know the factorisation of n'. Once the factorisation is known, one can then use Cornacchia's algorithm [Cor07].

When we have flexibility on the choice of n' (we will see in Theorem 5.19 that we can take any n' such that $N = n + n'$ is smooth with N-torsion accessible, e.g., N is powersmooth), the fastest way is to try several n' until we find n' a prime congruent to 1 modulo 4 (or possibly a product of small primes congruent to 1 modulo 4 time a prime congruent to 1 modulo 4).

There is a randomized algorithm in time $O(\log^2 n')$ to decompose n' as a sum of four squares [RS86; PT18].

5.3. **The HD representation.** We have seen in Example 5.16 how we can now use Theorem 5.12 to obtain a special form of Theorem 5.1. We can give a more precise complexity statement:

Theorem 5.19. *Let $\phi : E_1 \to E_2$ be an n-isogeny, $(P_1, Q_1, \dots, P_r, Q_r)$ a CRT basis of the N-torsion, with $N = \prod_{i=1}^m \ell_i^{e_i} > n$ coprime to n. Let $u = 1, 2, 4$ according to whether $N - n$ is a sum of $1, 2, 4$ squares. Let $(P_i, \phi(P_i), Q_i, \phi(Q_i))$ be interpolation data of ϕ on $E_1[N]$. Then ϕ can be efficiently embedded into a $2u$-dimensional N-isogeny Φ, which can be decomposed as a product of ℓ_i-isogenies in time $O(\sum_i e_i \ell_i^{2u}(d_i \log e_i + \sum_{j>i(d_i \vee d_j)}) = \widetilde{O}(m^2 d' e \ell^{2u})$ arithmetic operations over $\mathbb{F}_q$.*

Given a point $P \in E_1(\mathbb{F}_{q'})$ with $q' = q^{d_P}$, and the decomposed representation of Φ, evaluating P requires $O(\sum_i e_i \ell_i^{2u}(d_i \vee d_P)) = O(me\ell^{2u} d'')$ arithmetic operations over $\mathbb{F}_q$.

Proof. We use Proposition 5.15 to build an n'-endomorphism $\psi : E_1^u \to E_1^u$. We can then use Corollary 5.13 to ψ and $\phi\,\mathrm{Id} : E_1^u \to E_2^u$ to embed ϕ into a N-isogeny of dimension $2u$ $\Phi : E_1^u \times E_2^u \to E_1^u \times E_2^u$. (We remark that since ψ is a matrix of integers, it commutes with $\phi\,\mathrm{Id}$ and so $\psi' : E_2^u \to E_2^u$ is given by the same matrix as ψ.) Although Corollary 5.13 is stated for elliptic curves, by Theorem 5.12 which was stated for principally polarised abelian varieties, it extends to product of elliptic cuves (and their product principal polarisation).

Once we have Φ we can use the higher dimensional Vélu's like formula to decompose it. The complexity then follows from Lemma 5.7. See also [Rob22a; Rob22b] for more details. □

Remark 5.20.

- The complexity of the HD representation depends on the dimension $g = 2u$ of Φ, the isogeny we embed ϕ into, but also the largest prime divisor of N and the field of definition of the points of $E_1[\ell_i^{e_i}]$ torsion. In the best case, we have $g = 2$, $N = 2^e$, and $E_1[2^e]$ has rational points. We note that to achieve $g = 2$ we need to find a $n' = N - n$ dimension 1 isogeny $\psi : E_1 \to E_1'$, which can be hard (unless n' is a square). However, if we know $\mathrm{End}(E_1)$, it is much easier to build isogenies of appropriate degree. That's the main idea behind the algorithms of SQIsign2d [BDD+24; NO24; DF24], which build an efficient representation of the response isogeny by embedding it into a 2^e-isogeny in dimension 2.
- If N is smooth and the N-torsion on E_1 is accessible, but N is not coprime to n, we can write $N = dN_1, n = dn_1$ with $d = N \wedge n$. Then ϕ splits as $\phi = \phi_1 \circ \phi_0$ where $\phi_0 : E_1 \to E_{11}$ is a d-isogeny whose kernel is efficiently computed since $d \mid N$ and we know how ϕ acts on the N-torsion, and $\phi_1 : E_{11} \to E_2$ is a n_1-isogeny. We can then apply Theorem 5.19 to ϕ_1 to embed it into an N_1-isogeny.

Definition 5.21. The HD representation of ϕ can use any efficient representation of an higher dimensional isogeny Φ such that ϕ embeds into Φ. In practice, following Theorem 5.19, the HD representation of ϕ can consist:

(1) The interpolation data: $(P_i, \phi(P_i), Q_i, \phi(Q_i))$, with N smooth and $E_1[N]$-accessible. By Theorem 5.12, this is essentially equivalent to a multigenerator representation of $\ker \Phi$. We call this the torsion HD-representation.
(2) The decomposition of Φ as a product of small isogenies. This only requires N-smooth, but of course to convert from the torsion HD-representation into a decompose HD-representation we also need N accessible.

Corollary 5.22. *The HD representation is universal. Any other efficient representation of ϕ can be efficiently converted (meaning in polynomial time in $\log n$ and $\log q$) into an HD representation.*

Proof. It suffices to select a smooth integer N with accessible N-torsion (for instance N powersmooth), construct a CRT basis of the N-torsion, and use our efficient representation to evaluate ϕ on this CRT basis. We thus obtain a torsion HD representation. (The HD representation needs the degree of ϕ, but we have assumed that this is always part of the data to represent ϕ). □

As mentioned in the introduction, we can use Corollary 5.22 to show that we just need to be able to evaluate ϕ efficiently on enough nice points $P \in E(\mathbb{F}_{q'})$ to be able to efficiently evaluate ϕ on all points P, even in k-algebras.

Remark 5.23. Given an HD like data, say a CRT basis (P_i, Q_i) of $E_1[N]$, and points $(R_i, S_i) \in E_2[N]$ that are the putative images of P_i, Q_i by some isogeny ϕ, one can ask whether it really encodes some isogeny.

The HD representation takes the torsion information above, and convert it into the kernel of an N-isogeny $\Phi : A \to B$, where A, B are supposed to split as product of elliptic curves: $A = \prod E_i, B = \prod E_j'$. (The expected decomposition depends on the dimension used, and whether we used auxiliary endomorphisms or isogenies.)

If the codomain B of Φ, does not split as a product of elliptic curves, we know that Φ does not encodes a dimension one isogeny. When B splits, then Φ consists of a matrix of n_{ij}-isogenies $\phi_{ij} : E_i \to E_j'$, with $n_{ij} \leq N$ (we can recover n_{ij} using pairings, see Lemma 6.2). In that case, Φ does encode (several!) dimension one isogenies. To check whether it encodes

a particular candidate, i.e. whether a ϕ_{ij} is equal to some specific isogeny $\phi : E_1 \to E_2$, we can first look for a E'_j isomorphic to E_2, and then (if we know how ϕ is supposed to act on sufficiently many points) use the equality algorithm of Proposition 6.1 to check for equality.

6. Algorithms on efficient representation of isogenies

Using the universality of the HD representation (Corollary 5.22), we can build many algorithms on efficiently represented isogenies.

6.1. **Equality and sum.** We start with standard algorithms.

Proposition 6.1. *Given two efficient representations of two n-isogenies $\phi_1, \phi_2 : E_1 \to E_2$, we can efficiently test if ϕ_1 is equal to ϕ_2.*

Proof. Here we assume that the degree of ϕ_1, ϕ_2 are the same, otherwise they are clearly not equal (see Lemma 6.2 on how to recover the degrees if we only have a bound on it). We also assume that the codomain is the same; it can be useful to test equality up to postcomposition by an automorphism, but it suffices to apply Proposition 6.1 to all automorphisms.

To test equality, we simply construct a CRT basis of the N-torsion on E_1 for any $N > 2\sqrt{n}$ where the N-torsion is accessible. Then $\phi_1 = \phi_2$ if and only if they agree on this CRT basis. □

If we have an efficient representation of a n_1-isogeny $\phi_1 : E_1 \to E_2$ and of a n_2-isogeny $\phi_2 : E_2 \to E_3$ then we have an efficient representation of the $n_1 n_2$-isogeny $\phi_2 \circ \phi_1 : E_1 \to E_3$ (say by keeping it decomposed as ϕ_1 followed by ϕ_2).

If we have a n_1-isogeny $\phi_1 : E_1 \to E_2$ and a n_2-isogeny $\phi_2 : E_1 \to E_2$ that have an efficient representation, then of course we can also evaluate the sum $\phi_1 + \phi_2$ efficiently on any point. But a catch is that in our representation data, we require to know the degree of the isogeny; and Cauchy-Schwarz only gives us a bound on the degree of $\phi_1 + \phi_2$. This is in fact enough, thanks to pairings:

Lemma 6.2. *Let $\phi : E_1 \to E_2$ be an isogeny of degree at most n. Assume that we know the evaluation of ϕ on a CRT basis of the N-torsion, with N smooth and the N-torsion accessible, and $N > n$. Then we can efficiently recover the degree of ϕ.*

Proof. If $\deg \phi = d$, for $P, Q \in E[N]$, we have $e_N(\phi(P), \phi(Q)) = e_N(P, Q)^d$. We can thus recover d by computing Weil pairings and discrete logarithms (working separately on each prime power of N). □

Corollary 6.3. *If we have an efficient representation of two isogenies $\phi_1, \phi_2 : E_1 \to E_2$, we have an efficient representation of their sum.*

6.2. **Duals and divisions.** Now, we describe algorithms that need the HD representation. These are algorithms that are relatively straightforward, simply using the idea that the HD representation only needs to know the evaluation of ϕ on sufficiently many nice points to extract from it an HD representation.

Proposition 6.4. *If we have an efficient representation of an n-isogeny $\phi : E_1 \to E_2$, then we also have an efficient representation of the dual n-isogeny $\widetilde{\phi} : E_2 \to E_1$.*

Proof. Take a CRT basis (P_i, Q_i) of $E_1[N]$, with N large enough and prime to n, smooth and the N-torsion accessible. Then $(P'_i = \phi(P_i), Q'_i = \phi(Q_i))$ is a CRT basis of $E_2[N]$ (because N is prime to n), which can be efficiently evaluated by our assumption on ϕ. Now,

by definition of the contragredient isogeny, we have $\widetilde{\phi}(P'_i) = nP_i, \widetilde{\phi}(Q'_i) = nQ_i$. Thus we have an HD representation of $\widetilde{\phi}$.

(We remark that[11], if we are given a basis (P, Q) of $E_1[N]$ and the action of ϕ on this basis, and we are given a basis (P', Q') of $E_2[N]$, then using that $e_{W,N}(\phi(R), S) = e_{W,N}(R, \widetilde{\phi}S)$ where $R \in E_1[N]$ and $S \in E_2[N]$, we can recover how $\widetilde{\phi}$ acts on P', Q' via Weil pairings and dlp computations in μ_N, even if N is not coprime to n. This can be useful to reconstruct the action of $\widetilde{\phi}$ on some N-torsion without going all the way to an HD representation.) □

Example 6.5. If $E/\mathbb{F}_q$ is an elliptic curve, the Frobenius π_p can be efficiently computed: $\pi_p(x(P), y(P)) = (x^p(P), y^p(P))$. Hence its dual, the Verschiebung $\hat{\pi}_p$ can also be efficiently computed in $O(\log^{O(1)}(p))$. Computing the Verschiebung via its kernel (which consist of the étale points of p-torsion if E is ordinary) would take $O(p)$. If E is ordinary, evaluating the Verschiebung on differentials, we can recover the invertible eigenvalue of the Frobenius modulo p, hence its trace modulo p. This gives a polynomial point counting algorithm if $q = p$.

Let $\phi : E_1 \to E_2$ be an isogeny, and m an integer. The isogeny ϕ is divisible by m if and only if $\phi(E_1[m]) = 0$. If we have an efficient representation of ϕ and the m-torsion of E_1 is accessible, we can thus test divisibility by m efficiently. Using the HD representation, we can handle the general case:

Proposition 6.6. *If we have an efficient representation of an n-isogeny $\phi : E_1 \to E_2$, then we can test if ϕ is divisible by some integer m, and if so obtain an efficient representation of ϕ/m.*

Proof. Take a CRT basis (P_i, Q_i) of $E_1[N]$, with N large enough and prime to n, smooth and the N-torsion accessible. Then $(P_i, \phi(P_i)/m, Q_i, \phi(Q_i)/m)$ gives a torsion HD-representation of ϕ/m, if it exists.

To test if this putative HD data is valid for ϕ/m, we can use Remark 5.23, because we know how ϕ/m is supposed to act if it exists. □

Example 6.7. Assume that we have an efficient representation of an endomorphism $\alpha \in \mathrm{End}(E)$, and let $R = \mathbb{Q}[\alpha] \cap \mathrm{End}(E)$ be the saturation of $\mathbb{Z}[\alpha]$ in $\mathrm{End}(E)$. Assume that we know the factorisation of the conductor f_α of $\mathbb{Z}[\alpha]$ in its maximal order R_0. Then we can apply the ℓ-division algorithm to suitable endomorphisms on $\mathbb{Z}[\alpha]$ for $\ell \mid f_\alpha$ to recover the conductor f_R of R in R_0. In other words, we can recover in polynomial time (in $\log \Delta_\alpha$) the saturation of a quadratic order $\mathbb{Z}[\alpha]$ in $\mathrm{End}(E)$, provided we have a factorisation of the conductor f_α.

When E is ordinary, $\mathrm{End}(E)$ is the saturation of $\mathbb{Z}[\pi_q]$, and we can apply the above algorithm since π_q has an efficient representation. This gives a polynomial time algorithm to compute $\mathrm{End}(E)$, provided we have a factorisation of the conductor of $\mathbb{Z}[\pi_q]$. See [Rob22b] for more details, and [MW23; PW24; ES24] for other applications of the saturation algorithm.

Similarly, for any elliptic curve, if R is the saturation of $\mathbb{Z}[\pi]$ in $\mathrm{End}(E)$ (this saturation is the same whether we take $\mathrm{End}_{\mathbb{F}_q}(E)$ or $\mathrm{End}_{\overline{\mathbb{F}}_q}(E)$ for $\mathrm{End}(E)$), then each endomorphism $\alpha \in R$ admits an efficient representation, namely polynomial in $\log \Delta_\alpha$ and $\log q$. Indeed, we can write $\alpha = (a + b\pi)/f$, with f dividing the conductor of $\mathbb{Z}[\pi]$ (hence at most $O(q)$), and $\log|a|, \log|b|$ are in $O(\Delta_\alpha + \log q)$. We can evaluate $a + b\pi$ efficiently, hence α too by the division algorithm.

[11] This argument was communicated by Benjamin Wesolowski

Corollary 6.8. *Given efficient representations of $\phi : E_1 \to E_2$ and of $\phi_r : E_1' \to E_2$ and $\phi_l : E_1 \to E_2'$, one can efficiently test if $\phi = \phi_r' \circ \phi_l$ and if $\phi = \phi_r \circ \phi_l'$, and if so output an efficient representation of ϕ_r' and ϕ_l'.*

Proof. We treat the case of ϕ_l, the other case being symmetric. We have $\phi = \phi_r' \circ \phi_l$ if and only if $\phi \circ \widetilde{\phi_l} = \phi_r' \circ [\deg \phi_l]$, so, since we know an efficient representation of $\widetilde{\phi_l}$ by Proposition 6.4, the question reduces to the question of division by an integer, which is handled by Proposition 6.6. □

6.3. **Advanced algorithms on efficient representations.** In this section, we describe some algorithms that need to delve into how the HD representation works rather than treating it simply as a black box.

Proposition 6.9. *Given an efficient representation of $\phi : E_1 \to E_2$ over a finite field $\mathbb{F}_q$, and given $R = \mathbb{F}_q[\varepsilon]/\varepsilon^m$ or $R = \mathbb{Z}_q/p^m\mathbb{Z}_q$ and a deformation/lift $\widetilde{E_1}/R$ of E_1 to R, one can find an efficient representation of the isogeny ϕ deformed/lifted to R.*

Proof. The idea is as follows: we compute a decomposed HD representation $\Phi : A \to B$ of ϕ, where Φ is split into a product of small higher dimensional isogenies Φ_i. We lift ϕ by lifting Φ, which amount to lifting the kernel of each Φ_i.

There is however one technical difficulty with this approach. Let's say we use a 8-dimensional representation $\Phi : E_1^4 \times E_2^4 \to E_1^4 \to E_2^4$ of $\phi : E_1 \to E_2$. Then when we want to compute the deformation $\widetilde{\phi}$ of ϕ to $\widetilde{E_1}/R$, we don't yet know the codomain $\widetilde{E_2}$ of $\widetilde{\phi}$. So if $A = E_1^4 \times E_2^4$, we don't know which is the correct deformation $\widetilde{A}/R$ of A we need to embed $\widetilde{\phi}$!

The solution is to make an arbitrary choice $\widetilde{E_2}'$ for $\widetilde{E_2}$. If our choice is incorrect, we will get a $B = \widetilde{E_1}^4 \times (\widetilde{E_2}'')^4$ with $\widetilde{E_2}'' \neq \widetilde{E_2}'$. In other words, to get the correct deformation $\widetilde{\Phi}$, we want to find $\widetilde{E_2}'$ such that $\widetilde{E_2}''$ is equal to $\widetilde{E_2}'$. We can solve this by a Newton algorithm, where each steps reduces to a linear algebra problem on the deformation spaces of E_1 and E_2. The corresponding matrix can be computed by computing the $\widetilde{E_2}''$ associated to the $\widetilde{E_2}'$ for a basis of the deformation space. We refer to [Rob22b; KR24] for more details. □

Example 6.10. If $E/\mathbb{F}_q$ is ordinary, one can use Proposition 6.9 to compute the canonical lift $\widetilde{E}/\mathbb{Z}_q$ to p-adic precision m in polynomial time in $\log p$. Combined with Example 6.5, this gives a point counting algorithm in $O(n^2 \log^C p)$ when $q = p^n$ [Rob22b].

One can also use deformations of isogenies to $\mathbb{F}_q[\varepsilon]/\varepsilon^m$ to compute modular polynomials efficiently [KR24].

Proposition 6.11. *Assume that we have an efficient representation of an isogeny $\phi : E_1 \to E_2$ of degree $n = n_1 n_2$ with $n_1 \wedge n_2 = 1$. Then ϕ splits uniquely as $\phi = \phi_2' \circ \phi_1 = \phi_1' \circ \phi_2$ where ϕ_1, ϕ_1' are of degree n_1 and ϕ_2, ϕ_2' are of degree n_2. Furthermore, we can efficiently find an efficient representation of $\phi_1, \phi_2, \phi_1', \phi_2'$.*

Proof. If $K = \operatorname{Ker}\phi$, we can define ϕ_1 (resp. ϕ_2) to be the isogeny with kernel $K[n_1]$ (resp. $K[n_2]$) and ϕ_2' (resp. ϕ_1') to be the isogeny with kernel $\phi_1(K)$ (resp. $\phi_2(K)$). Then $\phi_1, \phi_2, \phi_1', \phi_2'$ form a (n_1, n_2)-isogeny diamond, hence embed into a N-isogeny Φ in dimension 2 with $N = n_1 + n_2$. Furthermore, we can recover the kernel of Φ from the action of ϕ on $E_1[N]$. This solves the problem if N is smooth and E_1, E_2 have accessible N-torsion: computing Φ gives all four isogenies $\phi_1, \phi_2, \phi_1', \phi_2'$ at once (we can distinguish ϕ_1 from ϕ_2 using pairings since they don't have the same degree).

For the general case, we pad ϕ with extra isogenies. Namely we consider $\phi' = \psi_2 \circ \phi \circ \psi_1$ where ψ_1 (resp. ψ_2) is an efficient isogeny of degree u (resp. v). The goal is to find such

isogenies ψ_1, ψ_2 such that $N = un_1 + vn_2$, with N smooth and the N-torsion accessible, so that we can apply the splitting algorithm above to split ϕ' into $\psi_2 \circ \phi'_2$ and $\phi_1 \circ \psi_1$. Then we can apply the division algorithm to recover ϕ'_2, ϕ_1. If we want to recover ϕ'_1, ϕ_2 instead, we need to search for u, v such that $N = un_2 + vn_1$.

A simple pigeonhole argument shows that for any $N > n_1 n_2$, we can find $u, v > 0$ such that $N = un_1 + vn_2$. We remark that if N is coprime to $n_1 n_2$ then automatically un_1 will be coprime to vn_2.

Once we have fixed u, v, we can always find ψ_1, ψ_2 by going to higher dimension if needed: namely we work with E_1^r, E_2^r with $r = 1, 2, 4$, and letting by abuse of notations $\phi : E_1^r \to E_2^r$ to be the diagonal isogeny of $\phi : E_1 \to E_2$.

For instance, if $u = u_1^2 + u_2^2 + u_3^2 + u_4^2$ we build a quaternion matrix for ψ_1, and then using the algorithm above in dimension 4 (so using a Φ in dimension 8), we recover $\phi_1 \circ \psi_1$. This means that we obtain ϕu_i for $i = 1, 2, 3, 4$, hence we can directly recover $\phi_1 \gcd(u_i)$. Thus we don't even need to apply the division algorithm to recover ϕ_1 in the cases where we can find u_i coprime. This includes all odd integers. However if u has 2-adic valuation e, then 2^f has to divide all the u_i for $f = \lfloor (e-1)/2 \rfloor$ (and conversely we can find a solution with $\gcd(u_i) = 2^f$). □

Example 6.12 (QFESTA)**.** The first application of splitting isogenies was given in QFESTA [NO23]. Namely we have seen in Example 5.14 that on $E_0 : y^2 = x^3 + x/\mathbb{F}_{p^2}$, with $p \equiv 3 \mod 4$, we can efficiently build endomorphisms of norm $D \gg p$. Suppose that we want to build an efficient isogeny of degree n. We first look for an endomorphism γ of norm $D = n(N - n)$ where N is chosen to be prime to n, and smooth with accessible N-torsion on E_0. Then a direct application of Proposition 6.11 allows to split γ into $\gamma = \gamma_2 \circ \gamma_1$ where γ_1 is a n-isogeny. Our D is chosen so that the splitting can be done directly in dimension 2 without any padding required for γ. To find γ, we require that D is large enough ($D \gg p$); but on the other hand for efficiency we also want to take $N = 2^e$ where $2^e \mid p \pm 1$ so that the 2^e-torsion is rational over $\mathbb{F}_{p^2}$, this already imposes $N < p$. For some applications, we might have only have available 2^e-torsion with 2^e significantly smaller than p, making the condition $D \gg p$ hard to satisfy. A solution is to simply iterate the splitting: we take $D_1 = n(N_1 - n)$, $D_2 = D_1(N_2 - D_1)$, $D_3 = D_2(N_3 - D_2)$ for appropriately chosen divisors $N_1, N_2, N_3 \ldots$ of N, until we can find an endomorphism γ_i of norm D_i. Then we successively split γ_i to obtain an isogeny γ_{i-1} of degree D_{i-1}, then γ_{i-2} of degree D_{i-2} until we find γ_1 of degree n.

In [NO24], Nakagawa and Onuki extend this construction to build an isogeny of degree n from an arbitrary supersingular elliptic curve E, provided that the connecting ideal I between E_0 and E is known. The idea is to sample γ in the Eichler order of E_0 and E, using the tools introduced for the original SQIsign scheme (see [Ler22b] for a nice overview of these tools).

Example 6.13 (Clapotis)**.** The full power of the splitting algorithm of Proposition 6.11 was introduced for the Clapotis group action in [PR23b]. We want to compute the isogeny $\phi_I : E_1 \to E_2$ associated to an *invertible* class group ideal I, provided that we know an efficient representation of $\mathrm{End}(E_1)$. It suffices to find two equivalent ideals J_1, J_2 to I of coprime degree. Then $\phi = \widetilde{\phi_{J_2}} \circ \phi_{J_1}$ is an endomorphism of E_1, which we know how to efficiently evaluate. We can thus apply Proposition 6.11 to recover an efficient representation of ϕ_{J_1}, ϕ_{J_2}; this build a double path from $E_1 \to E_2$ from which it is easy to get an efficient representation of ϕ_I too (but for the class group action usually we only need to recover the codomain E_2 rather than the full ϕ_I).

The same idea apply to convert an ideal to isogeny in the supersingular setting, and is the basis of [BDD+24] (see Appendix C for more details).

Example 6.14. Let $E_1/\mathbb{F}_q$ be an elliptic curve of cardinal divisible by a prime ℓ, and assume that $\#E(\mathbb{F}_q)$ is not divisible by ℓ^2. Let $K = \langle T \rangle$ be the kernel generated by a point $T \in E_1[\ell](\mathbb{F}_q)$. Then we can efficiently compute $\phi : E_1 \to E_2 = E_1/K$. Indeed, $\pi_q - 1$ is efficient (see Example 6.7) of degree $\#E(\mathbb{F}_q)$, and we can recover ϕ by splitting it.

Similarly, assume that $\chi_\pi \mod \ell$ splits and is separable: $\chi_\pi(X) = (X - \lambda_1)(X - \lambda_2)$ modulo ℓ with $\lambda_1 \neq \lambda_2$; in other words the Frobenius has two distincts eigenvalues. Then ℓ split in $\mathbb{Z}[\pi]$, hence splits in R as $\ell = \mathfrak{l}_1\mathfrak{l}_2$, the saturation of $\mathbb{Z}[\pi]$ in $\mathrm{End}_{\mathbb{F}_q}(E_1)$, because from our assumptions ℓ does not divide the conductor of $\mathbb{Z}[\pi]$. In that case there are only two rational kernels K in $E_1[\ell]$ (given by the eigenvectors of λ_i), and we can apply Example 6.13 to compute the corresponding isogenies efficiently. The preceding example is a special case of this when $\lambda_1 = 1$ and $\lambda_2 = q \neq 1 \mod \ell$.

When $\lambda_1 = \lambda_2 = \lambda$, but π_q is not diagonal, then $\pi_q = \begin{pmatrix} \lambda & 1 \\ 0 & \lambda \end{pmatrix}$, and there is a unique rational kernel K in $E_1[\ell]$. Either E_1 is at the top of the volcano and $\ell = \mathfrak{l}^2$ is ramified in R, or we are at the bottom of the volcano: $R = \mathbb{Z}[\pi_q]$ and K is the kernel of the unique ascending isogeny (see Example A.4). In the first case we can apply Example 6.13 to compute efficiently the associated isogeny, but not in the second case because the ideal associated to K is not invertible.

Proposition 6.15. *Assume that we have an efficient representation of a n_1-isogeny $\phi_1 : E_0 \to E_1$ of degree n_1 and of a n_2-isogeny $\phi_2 : E_0 \to E_2$ with $n_1 \wedge n_2 = 1$. Then we have a pushforward square, with $\phi_2' : E_1 \to E_{12}$ the pushforward of ϕ_2 by ϕ_1, and $\phi_1' : E_2 \to E_{12}$ the pushforward of ϕ_1 by ϕ_2. And we can efficiently find an efficient representation of ϕ_1', ϕ_2'.*

Proof. Let $\phi = \phi_2 \circ \widehat{\phi_1} : E_1 \to E_2$, which admits an efficient representation by Proposition 6.4. Then we can split ϕ as $\phi = \phi_1'' \circ \phi_2''$ by Proposition 6.11, and we have $\phi_2' = \phi_2''$, $\phi_1' = \widehat{\phi_1''}$. □

Example 6.16. In the setting of Proposition 6.15, if ϕ_2 has smooth accessible kernel $K_2 = \mathrm{Ker}\,\phi_2$, then the pushforward ϕ_2' of ϕ_2 by ϕ_1 is given by the kernel $K_2' = \phi_1(K_2)$ (even if n_1 is not coprime to n_2). So we can directly compute ϕ_2' from its kernel K_2'.

Furthermore, since $\phi_1' \circ \phi_2 = \phi_2' \circ \phi_1$, we can find the evaluation of ϕ_1' on nice points, hence we can easily build an HD representation of ϕ_1'.

Thus, pushforwards in an hybrid setting, where ϕ_1 is given by an HD representation and ϕ_2 by a smooth accessible kernel allows for easier pushfowards; this is exploited to great effect in the POKE framework of [Bas24].

Example 6.17. Proposition 6.15 allows to generalise the "SIDH proof of knowledge" of isogenies from [DDGZ22; BCC+23; GPV24] from smooth isogenies to arbitrary efficient isogeny representations.

Namely, to prove the knowledge of an efficient n-isogeny $\phi : E_1 \to E_2$, provided that we have an algorithm to sample efficient isogenies of large degrees (coprime to $\deg \phi$) from E_1 (for instance because we know its endomorphism ring), one could build a pushforward square:

$$\begin{array}{ccc} E_1 & \xrightarrow{\phi} & E_2 \\ \downarrow{\scriptstyle \psi} & & \downarrow{\scriptstyle \psi'} \\ E_1' & \xrightarrow{\phi'} & E_2' \end{array}$$

and commit the efficient representations of ψ, ϕ', ψ', where the pushforwards ψ', ϕ' are computed by Proposition 6.15. Then the verifier ask to reveal one out of the three of ψ, ϕ', ψ', verify it encodes a valid isogeny representation, and this protocol is repeated sufficiently many times.

The soundness associated to this protocol is that the Prover knows some isogeny $E_1 \to E_2$. The Zero-Knowledge property is harder, since revealing ϕ' could leak informations on ϕ, that's why ψ is required to be of large enough degree.

A subtlety appear when the prover wants to prove that he knows an isogeny $\phi : E_1 \to E_2$ of explicit degree $n = n_1$. Even if he publicly commit to the degrees n_2 of ψ, ψ' and n_1 of ϕ' (which the verifier can check if the corresponding isogeny is revealed), he can only prove that he knows an isogeny $\widehat{\psi'} \circ \phi' \circ \psi$ of degree $n_1 n_2^2$.

Indeed, the verification process does not allow to prove that ψ' is the pushforward of ψ by ϕ (or equivalently, that ϕ' is the pushforward of ϕ through ψ, or equivalently that $\psi' \circ \phi = \phi' \circ \psi$, or equivalently that $\widehat{\psi'} \circ \phi' \circ \psi$ is divisible by $[n_2]$).

In the SIDH proof of isogeny knowledge, to prove the degree of ϕ, the authors of [DDGZ22] explain how the prover can commit informations about the kernels of ψ, ψ' (or more precisely their duals) to convince the verifier that they form a pushforward square. See also the survey [BDGP23] and [DFP24, § 5.5]. However, their solution requires that if n_2 is the degree of ϕ, then n_2 should be smooth and the n_2-torsion accessible on E_1, E_2, E_1', E_2'. So we cannot use an arbitrary efficient isogeny for ψ anymore.

Still, the method of [DDGZ22] only require to be able to construct the pushforward ϕ' and to evaluate it on points, so by Proposition 6.15 it still works for any efficient representation of ϕ. In fact, since this proof of knowledge with explicit degree requires taking ψ to be smooth with accessible torsion, we could simply use Example 6.16 directly instead of Proposition 6.15. We remark also that, using Proposition 6.15, we can generalize the method of [DDGZ22] to any ψ such that the n_2-torsion is accessible on E_1', E_2', provided that we can find an efficient representation of ψ (since we are relaxing the constraint n_2-smooth, we cannot use the decomposed representation anymore). Although this constraint on ψ is not too big in practice, as far as we know it remains an open question to handle the case of an arbitrary efficient representation for ψ, where the n_2-torsion might not be accessible. Maybe a solution would be to use many distinct degrees n_2 for ψ: if the prover can convince the verifier that he knows isogenies of degree $n_1 n_2^2$ between E_1 and E_2 for sufficiently many distinct degrees n_2, then maybe this is enough to prove that he knows an isogeny of degree n_2, because the quadratic lattice $(\mathrm{Hom}(E_1, E_2), \deg)$ is of rank at most four.

6.4. **Kernel.** Given an efficient representation of a cyclic n-isogeny $\phi : E_1 \to E_2$, one can ask whether we can recover its kernel. If the n-torsion is accessible on E_1, we can evaluate ϕ on a basis of the n-torsion, and solve a DLP in E_2 (or in μ_n if the n-torsion is also accessible on E_2 so that we use the Weil pairing) to recover $\ker \phi$. This is efficient if n is smooth.

If the n-torsion is accessible on E_2, we can compute a basis (R, S) of $E_2[n]$, and compute $(P, Q) = (\widehat{\phi}(R), \widehat{\phi}(S))$ by Proposition 6.4. Then (P, Q) is a multigenerator representation of $\ker \phi$. If we know the factorisation of n, we can even compute the orders of P, Q and from that extract a generator. This can allow to relax the condition that n is smooth.

In all cases, we can at least recover the kernel in time $\widetilde{O}(n)$:

Lemma 6.18. *If we have an efficient representation of an n-isogeny $\phi : E_1 \to E_2$, we can recover its function representation and an equation for its kernel in $\widetilde{O}(n)$ arithmetic operations.*

Proof. Since we know how to evaluate ϕ on every point, in principle it would be enough to evaluate it on the generic point $\mathbf{P} \in E_1(k(E_1))$ of E_1. The problem is that $k(E_1)$ is

a rational function field over k, and the cost on the arithmetic on $k(E_1)$ depends on the degrees of the intermediate functions in $k(E_1)$ computed during the evaluation. To bound these intermediate degrees, a solution is to work with the formal group to some precision $m = \Theta(n)$, and do a rational reconstruction at the end. But we have seen in Section 3.3 that it is enough to work in precision $m = 2$ to obtain the deformation representation, from which we can reconstruct the isogeny. (An alternative strategy would be to try to interpolate from the evaluation on several rational points).

Let us detail this formal group strategy. We evaluate ϕ on a fat point $\mathbf{P} \in E_1(k[\varepsilon]/\varepsilon^2)$ to recover the action of ϕ on differentials, hence recover the deformation representation. We then solve a differential equation in $\widetilde{O}(n)$ to recover the function representation, from which we can extract the kernel. This assumes that $p > 8n - 5$. For large n, we just need to lift E_1 arbitrarily to $\mathbb{Z}_q/p^m\mathbb{Z}_q$ with $m = O(\log n)$ and then lift ϕ; we know how to do that efficiently thanks to Proposition 6.9. □

7. Open questions

Although the toolbox to manipulate efficient representations of isogenies has considerably expended, there are still many open questions.

We discuss some of these here:

(1) The most important one is to find in polynomial time in $\log n$ an efficient representation of an n-isogeny $\phi : E_1 \to E_2$ represented by a rational generator $\mathrm{Ker}\,\phi = \langle T \rangle$, $T \in E_1[n](\mathbb{F}_q)$. The best available algorithm (if n is not smooth, e.g. when n is a large prime), `sqrtVelu`, takes "exponential" time $\widetilde{O}(n^{1/2})$.

(2) If $\phi : E_1 \to E_1$ is a cyclic ℓ^2-isogeny, then it decomposes uniquely as $\phi = \phi_2' \circ \phi_1$ with ϕ_1, ϕ_2' ℓ-isogenies. But Proposition 6.11 needs the coprimality condition to split ϕ efficiently. Thus, assuming that we have an efficient representation of ϕ, computing (efficient representations of) ϕ_2', ϕ_1 efficiently remains an open question.

(3) We have the same question about computing the pushforward of isogenies ϕ_1, ϕ_2 whose degree are not coprime.

In certain cases, we can give some answers. If $\phi_1 : E_0 \to E_1$ is an ℓ-isogeny and $\phi_2 : E_0 \to E_2$ is also an ℓ-isogeny, with ℓ prime, then either they have the same kernel in which case their pushforwards are given by an isomorphism $E_1 \simeq E_2$, or they have disjoint kernel in which case their pushforwards are $\widetilde{\phi_1}$ and $\widetilde{\phi_2}$. Thus, when ϕ_1 is a n_1 isogeny and ϕ_2 is a n_2-isogeny, we can still compute their pushfoward in some cases even where n_1 is not coprime to n_2. Namely we split them into isogenies of coprimary degree, and use pushout squares to reduce to the case $n_1 = \ell^{e_1}$ and $n_2 = \ell^{e_2}$. We have seen how to handle the case $e_1 \leq 1$ and $e_2 \leq 1$. We can assume $e_1 \leq e_2$. We can use Corollary 6.8 to test if ϕ_2 is divisible by ϕ_1: $\phi_2 = \phi_2' \circ \phi_1$, in which case the pushforward of ϕ_1 is the identity and the pushforward of ϕ_2 is ϕ_2'. But to handle the general case, we would need to know how to split ϕ_1, ϕ_2 into chunks of ℓ-isogenies. Thus the pushforward question reduces to the splitting question.

(4) Given a supersingular elliptic curve $E/\mathbb{F}_{p^2}$, it admits rational isogenies of any degree. Can we efficiently construct one of fixed arbitrary degree n (or even random large degree)? Currently we only know how to do that for n smooth (by taking a product of small isogenies) or when we know $\mathrm{End}(E)$ (by using an IdealToIsogeny algorithm).

(5) Can we climb up in a R-oriented ℓ-isogeny volcano when ℓ is large, i.e., can we find an efficient way to compute an ascending isogeny? We do not know how to compute the climbing up isogeny efficiently, even if we already know the codomain

(but see [Gal24] for a speed up when the codomain is known). Solving the climbing up problem with known codomain would have important implications on the post-quantum security of SCALLOP and SCALLOP-HD, due to [CII+23].

For the climbing up isogeny, we do have a compact representation given by a non invertible ideal. Indeed, the conductor ideal $\mathfrak{f} = \mathbb{Z} + fR_0 \subset R$ of $R = \mathrm{End}(E)$ encodes the isogeny ascending all the way to the top, so $(\ell, \mathfrak{f})$ is the ideal representing one step up. In that case, due to the non invertibility, Clapoti(s) does not apply and we do not now how to convert this ideal to an isogeny, other than by computing it from its kernel, which takes time $\widetilde{O}(\ell^2 \log^2 q)$. But see [Gal24] again for some improvements to this naive method in some cases.

Solving the ℓ^2-splitting problem would give an algorithm to climb: climbing up from E_1 to E_2 and then descending back down to some E_1' corresponds to an invertible ideal $\mathfrak{a}$ of norm ℓ^2 in $\mathrm{Cl}(\mathrm{End}(E_1))$, and thanks to Clapotis we know how to convert $\mathfrak{a}$ into an ℓ^2-isogeny ϕ. Splitting ϕ would then give us E_2.

(6) The same question holds for a descending isogeny, but in that case we do not even have an ideal representation.

(7) Can we find an improvement to the KLPT algorithm that produces powersmooth ideals of norm $\approx p$ rather than $\approx p^{4.5}$ (see Appendix C.5)?

Finally, as we have illustrated several times, having a theoretical polynomial time algorithm is often not good enough: for isogeny based cryptography we want fast algorithms. In practice, this means finding elliptic curves with large accessible 2^e-torsion, and embedding isogenies into higher dimensional 2^e-isogenies of dimension r, with r as small as possible (and ideally $r = 2$). So even through the algorithms in Section 6 are polynomial time, there are probably many tricks still open to speed them up.

Appendix A. On the accessible N-torsion of an abelian variety

In this section, we define the notion of accessible torsion.

Definition A.1. The N-torsion of an elliptic curve $E/\mathbb{F}_q$ is said to be *accessible* if we can work with points of N-torsion by using field extensions of small degree (meaning polynomial in $\log N$).

In practice, this means that if $N = \prod \ell_i^{e_i}$, we want to find CRT generators $(P_i, Q_i) \in E[\ell_i^{e_i}]$ which live in extensions of (uniform) degree $O(\log^{O(1)} N)$.

This definition extends to an abelian variety (and to subgroups G): the N-torsion (resp. G-torsion) on $A/\mathbb{F}_q$ is accessible if we can find CRT generators $(P_1, \dots, P_g) \in A[\ell_i^{e_i}]$ (resp. $G[\ell_i^\infty]$) which live in extensions of (uniform) degree $O(\log^{O(1)} N)$ (resp. $O(\log^{O(1)} \#G)$).

Remark A.2. We remark that while the (P_i, Q_i) are assumed to live in small extensions, the basis of N-torsion $P = \sum P_i, Q = \sum Q_i$ could be defined over an extension of large degree. So to work with accessible N-torsion, we need to work with the CRT basis directly.

Sometimes it will be useful to be able to work with linear combinations of the points (P_i, Q_i, P_j, Q_j). Let d_i (resp. d_j) be the degree of the field extension where (P_i, Q_i) (resp. (P_j, Q_j)) are defined.

If d_i is coprime to d_j, then one can construct the compositum field of degree $d_i d_j$ in quasi-linear time $\widetilde{O}(d_i d_j)$ along with embeddings $\mathbb{F}_{q^{d_i}} \to \mathbb{F}_{q^{d_i d_j}}$ and $\mathbb{F}_{q^{d_j}} \to \mathbb{F}_{q^{d_i d_j}}$ in quasi-linear time $\widetilde{O}(d_i d_j)$ too [DDS14]. Switching between the bivariate representation $\mathbb{F}_{q^{d_i d_j}} = \mathbb{F}_q[X, Y]/\langle \Phi_{d_i}(X), \Phi_{d_j}(Y) \rangle$ and the univariate representation of $\mathbb{F}_{q^{d_i d_j}}$, where

$\Phi_{d_i}(X)$ (resp. $\Phi_{d_j}(Y)$) are irreducible defining polynomials for $\mathbb{F}_{q^{d_i}}$ (resp. $\mathbb{F}_{q^{d_j}}$), can be done by modular composition, which can be computed in pseudo-linear time in the bit model by [KU11].

In the general case, if $d' = d_i \wedge d_j$, we can construct the subfield $\mathbb{F}_{q^{d'}} \subset \mathbb{F}_{q^{d_i}}$ by taking a random element $\alpha \in \mathbb{F}_{q^{d_i}}$, applying the trace $\beta = \mathrm{Tr}_{\mathbb{F}_{q^{d_i}}/\mathbb{F}_{q^{d'}}}(\alpha)$, and computing the minimal polynomial of β by power projection [Sho99] which can be done in pseudo-linear time by the transposition principle. Doing the same to construct $\mathbb{F}'_{q^{d'}} \subset \mathbb{F}_{q^{d_j}}$, we can find an isomorphism $\mathbb{F}'_{q^{d'}} \simeq \mathbb{F}_{q^{d'}}$ by computing the factorisation of the degree d' polynomial defining $\mathbb{F}'_{q^{d'}}$ over $\mathbb{F}_{q^{d'}}$; this costs $\widetilde{O}(d'^3 \log^2 q)$. For better algorithms for the isomorphism problem, we refer to [BDDFS19]. Evaluating the isomorphism is then again done by modular composition.

To simplify the complexity statements when decomposing an N-isogeny into product of ℓ_i-isogenies using accessible N-torsion, we will always suppose that we have precomputed the embedding of P_i, Q_i, P_j, Q_j into their common overfield $\mathbb{F}_{q^{d_i \vee d_j}}$. And similarly to push a point P defined over $\mathbb{F}_{q^{d_P}}$, we will assume that we have precomputed the embedding of P, P_i, Q_i into $\mathbb{F}_{q^{d \vee d_i}}$. (This last point is only needed if we represent the small isogenies by the generator of their kernel rather than by a kernel equation.) This step can be done in pseudo-linear time using fast modular composition.

To know whether the N-torsion is accessible, we can look at the smallest degree where the points of $\ell_i^{e_i}$ torsion on an elliptic curve E are defined. This is well known, and we summarize this in the following lemma (which we state for abelian varieties since this case is not harder than for elliptic curves).

Lemma A.3. *Let $A/\mathbb{F}_q$ be an abelian variety, and $\ell \neq p$ a prime. The minimal degree d_0 such that all points of ℓ-torsion in A are defined satisfy $d_0 \leq \ell^{2g} - 1$. Let $e > 0$ be the highest integer such that all points of ℓ^e-torsion of A are defined over $\mathbb{F}_{q^{d_0}}$. Then the minimal degree d such that all points of ℓ^{e+f}-torsion of A are defined is $d = d_0 \ell^f$.*

Proof. Let $\chi_\pi(X)$ be the characteristic polynomial of the Frobenius, this is a monic polynomial of degree $2g$. The degree d_0 is the minimal integer such that $\pi_q^{d_0} - 1 \equiv 0 \mod \ell$. The minimal polynomial $\Phi_{\pi,\ell}$ of the Frobenius acting on the ℓ-torsion $A[\ell]$ divides $\chi_\pi \mod \ell$, hence the order d_0 of π_q modulo ℓ is at most $\ell^{2g} - 1$.

Now we look at the action of π_q on the Tate module $T_\ell A$. By assumption, we have that $\pi_q^{d_0} - 1$ is of ℓ-adic valuation e: $\pi_q^{d_0} = 1 + \ell^e P + O(\ell^{e+1})$. Then $\pi_q^{d_0 r} = 1 + r\ell^e P + O(\ell^{e+1})$, so $\pi_q^{d_0 r} - 1$ is of ℓ-adic valuation at least $e+1$ if and only if $\ell \mid r$. It follows that the ℓ^{e+1}-torsion is defined over $\mathbb{F}_q^{d_0 \ell}$, and not over a smaller subfield. We conclude by recurrence. □

Example A.4. Let $\chi_\pi = X^2 - tX + q$ be the characteristic polynomial of the Frobenius π_q acting on an elliptic curve $E/\mathbb{F}_q$, then $\chi_\pi \mod \ell$ is the characteristic polynomial of π_q acting on $E[\ell]$ ($\ell \neq p$).

- If this polynomial is irreducible modulo ℓ, then we have two distinct eigenvalues λ_1, λ_2 of π_q defined over $\mathbb{F}_\ell^2$. We let d_i be the order of λ_i, and d_q the order of q modulo ℓ (this is called the embedding degree in pairing based cryptography). There are no rational cyclic kernel in $E[\ell]$, and the ℓ-torsion becomes defined over an

extension of degree $d_1 \vee d_q = d_2 \vee d_q$. In that case the ℓ-isogeny volcano is of height 1: the conductor of $\mathbb{Z}[\pi]$ is not divisible by ℓ, and in particular if R is the saturation of $\mathbb{Z}[\pi]$ in $\mathrm{End}(E)$, the index $[R : \mathbb{Z}[\pi]]$ is not divisible by ℓ. Furthermore, ℓ is inert in $\mathbb{Z}[\pi]$ and R.

- If $\pi_q = (X - \lambda_1)(X - \lambda_2) \mod \ell$ splits with no multiplicity, we have two distinct eigenvalues $\lambda_1, \lambda_2 \in \mathbb{F}_\ell$, we have two rational kernels (formed by the eigenvectors for λ_i), whose points are defined over an extension of degree d_1 and d_2 respectively. The full ℓ-torsion is defined over $d_1 \vee d_q = d_2 \vee d_q$. Like in the above example the ℓ-isogeny volcano is of height 1, but in this case ℓ splits in $\mathbb{Z}[\pi]$ and R, and the two kernels corresponds to the two ideals $\mathfrak{l}_i$ where $\ell = \mathfrak{l}_1\mathfrak{l}_2$ in R.
- If $\pi_q = (X - \lambda)^2 \mod \ell$, then either $\pi_q = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}$ or $\pi_q = \begin{pmatrix} \lambda & 1 \\ 0 & \lambda \end{pmatrix}$. We can distinguish these two cases by testing whether $\pi - \lambda$ is divisible by ℓ, this is a special case of the endomorphism ring algorithm of Example 6.7.

  In the first case all kernels are rational, and in the second case only one is rational. If d is the order of λ, then since $\lambda^2 = q$ we have that $d = d_q$ or $d = 2d_q$. In the first case the full ℓ-torsion is defined over the extension of degree d, while in the second case over the extension of degree ℓd.

  In the first case, we have descending isogenies so we are not at the bottom of the volcano. In the second case, either the isogeny volcano is of height 1, $\ell = \mathfrak{l}^2$ is ramified in R (and $\mathbb{Z}[\pi]$), and the unique kernel is $K = E[\mathfrak{l}]$, with $\mathfrak{l}$ inversible. Or the isogeny volcano is of height > 1, we are on the bottom of the ℓ-isogeny volcano, and $K = E[\mathfrak{l}]$ where $\mathfrak{l}$ is the unique ideal of norm ℓ in R (necessarily non invertible in R whose conductor is divisible by ℓ). Climbing up via K, the Frobenius become diagonal on the codomain.

To generate a basis of the N-torsion of an elliptic curve, there are several methods.

- The simplest is to factorize the division polynomial ψ_N; this is a polynomial of degree $O(N^2)$. Using [KU11], this costs $\widetilde{O}(N^3)$.
- If the N-torsion lives in a small extension d, a more efficient way is to sample points in $E(\mathbb{F}_{q^d})$ and then multiply by appropriate cofactors to get points of N-torsion.

  This can be more tricky than it looks if we want a basis rather than simply points of N-torsion: imagine that $N = \ell$ is a large prime and that $E(\mathbb{F}_{q^d})[\ell^\infty] = \langle P_1, P_2 \rangle$ with P_1 of order ℓ^2 and P_2 of order ℓ. Then sampling a random point in $E(\mathbb{F}_{q^d})$ and multiplying by the cofactor, we get a random linear combination $P = aP_1 + bP_2$ which is of order ℓ^2 (unless by luck $\ell \mid a$), and so to get a point of ℓ torsion we need to look at ℓP which is a multiple of ℓP_1. So the naive algorithm only produce points in $\langle \ell P_1 \rangle$ with overwhelming probability.

  A general solution to this problem is given in [Sut11], for a specific solution on elliptic curves using the Weil or Tate pairing to speed up this computation, see [Rob21, § 5.6.2]. Using the Tate pairing and Pollard's rho to solve the DLP in $\mu_N \subset \mathbb{F}_{q^d}$, the complexity of this method is bounded by $\widetilde{O}(d^2 \log^2 q + \sqrt{N} d \log q)$.

  Rather than multiplying by a cofactor (an integer), it can be helpful to multiply our random point by a suitable endomorphism, see [Cou09; BGDS23].
- In supersingular isogeny based cryptography, we work with elliptic curves $E/\mathbb{F}_{p^2}$ such that $E(\mathbb{F}_{p^2}) = (\mathbb{Z}/p \pm 1)^2$ (depending on which quadratic twist we take).

So it is very easy to find p such that the N-torsion is accessible (and even rational over $\mathbb{F}_{p^2}$): simply take N such that $N \mid p+1$ or $N \mid p-1$. One can even split N as $N = N_1 N_2$ where $N_1 \mid p+1$ and $N_2 \mid p+2$, and work on E or its quadratic twist E' whenever appropriate (in practice we work with their Kummer line, which does not depend on the twist, over $\mathbb{F}_{p^2}$). This is enough as long as we can treat points of N_1-torsion independently from the points of N_2-torsion (otherwise we need to go to $\mathbb{F}_{p^4}$).

Furthermore, to generate a basis of the N-torsion it suffice to sample random points and multiply by the appropriate cofactor: the problem mentioned above does not appear in that case. We can use the Weil pairing to test when we have found generators. Thus we do not need DLPs, and the complexity is bounded by $\widetilde{O}(d^2 \log^2 q)$.

For the special case when $N = 2^e$, efficient algorithms to sample a deterministic basis of the 2^e-torsion have been developed using the Tate pairing (this is often called an entangled basis in the literature), see for instance [CJL+17; ZSPDB18; CEMR24, § 5.1].

Example A.5 (The kernel of an isogeny represented by an ideal I)**.** If a cyclic n-isogeny ϕ is represented by an ideal I, then both in the oriented and supersingular cases we have $\operatorname{Ker}\phi = E[I] \subset E[N(I)]$. We will assume that we have an efficient representation of the order associated to I. To find a multigenerator representation of the kernel, we have two methods. First we factorize $N(I) = \prod \ell_i^{e_i}$, then a CRT basis of $E[I]$ consists of finding a generator P_i of each $E[I, \ell_i^{e_i}]$, so we can reduce to the case $n = N(I)$ a prime power. We can write $I = (N(I), \alpha)$.

- The first idea is to work over the field $\mathbb{F}_{q^d}$ where the points of $E[I]$ are defined: $E[I](\overline{\mathbb{F}}_q) \subset E[N(I)](\mathbb{F}_{q^d})$. We sample generators of $E[N(I)](\mathbb{F}_{q^d})$ using the methods above, and then eventually solve a DLP (combined with pairings) to find the kernel of α.
- The second idea, used in [EPSV, § 4.1], relies on selecting α such that $N(\alpha) \wedge N(I)^2 = N(I)$, then $E[I] = \overline{\alpha}(E[N(I)])$. So sampling a basis of $E[N(I)]$ and applying $\overline{\alpha}$ to this basis, we recover generators for $E[I]$.

Appendix B. Relaxing the torsion requirement for the HD representation

In this section, we explain how splitting the HD isogeny Φ in two can allow to reduce the torsion information we need on ϕ to recover Φ.

B.1. **Splitting a smooth cyclic isogeny in two.** Let $\phi : E_1 \to E_2$ be a cyclic n-isogeny with kernel K, and assume that $n = n_1 n_2$. Then we can split ϕ as $\phi = E_1 \to^{\phi_1} E_{12} \to^{\phi_2} E_2$ where $K_1 = \operatorname{Ker}\phi_1 = K[n_1]$, and $\operatorname{Ker}\phi_2 = \phi_1(K)$.

If ϕ is smooth and the kernel K is accessible, we can efficiently find these kernels (see Proposition 6.11 for the general case, but the general case needs n_1 coprime to n_2)). It is convenient to work with $\widetilde{\phi_2}$ instead, whose kernel is given by $K_2 = \phi(E_1[n])$.

From the kernels K_1, K_2 of $\phi_1, \widetilde{\phi_2}$, we can recover ϕ: we compute a decomposed representation of $\phi_1 : E_1 \to E_{12}$ from K_1, $\widetilde{\phi_2} : E_2 \to E'_{12}$ from K_2, and we glue in the middle, i.e., compute an isomorphism $E_{12} \simeq E'_{12}$. Then we just need to compute the dual of the decomposed representation of $\widetilde{\phi_2}$ to obtain a way to evaluate ϕ. In other words: $(K_1 \subset E_1, K_2 \subset E_2)$ gives a representation of ϕ (the *split representation*), which is efficient if n is smooth, K_1 is accessible in E_1, and K_2 accessible in E_2.

For a supersingular curve over $\mathbb{F}_{p^2}$, since the Frobenius is $\pi_q = \pm[p]$ acts like a scalar on the n-torsion, and furthermore all supersingular curves have the same torsion structure, asking for K to be accessible is the same as asking the n-torsion to be accessible. In particular, the split representation relax the conditions on accessibility: if the n-torsion is accessible then certainly the n_1 and n_2-torsion are, but not conversely.

We will apply the same strategy to Φ in higher dimension for the HD representation.

B.2. **Splitting the HD representation in two.** In Section 5.3, we embed the n-isogeny $\phi : E_1 \to E_2$ into a higher dimensional N-isogeny $\Phi : A = E_1^u \times E_2^u \to B = E_1^u \times E_2^u$ of dimension $g = 2u$ by using an auxiliary $N - n$-endomorphism α on E_1^u given by a suitable matrix of integers, where $u = 1, 2, 4$ depending on whether $N - n$ is a sum of $1, 2, 4$ squares. As usual, we assume N smooth and the N-torsion accessible.

To recover the full kernel K of Φ, we need to know how ϕ acts on the N-torsion. But in fact, we need less information, because K is of rank g. First we remark that if we know how ϕ acts on the N-torsion, then we also have the full kernel of $K' = \widetilde{\Phi} = \Phi(A[N])$. If we write $N = N_1 N_2$, and we only require to know how ϕ acts on the $N_1 \vee N_2$-torsion, we can still recover $K[N_1]$ and $K'[N_2] = \Phi(A[N_2])$. This is enough to compute Φ: we decompose Φ as $\Phi = \Phi_2 \circ \Phi_1$, a N_1-isogeny followed by a N_2-isogeny. The kernel of Φ_1 is given by $K[N_1]$, which is indeed of degree N_1^g since K is of rank g, while the kernel of $\widetilde{\Phi_2}$ is given by $K'[N_2]$. So we can compute $\Phi_1 : A \to C$ and $\widetilde{\Phi_2} : C \to A$, and glue together in the middle at C to reconstruct Φ.

In summary, we just need to know how ϕ acts on the N-torsion, with $N^2 > n$, to be able to efficiently represent it via an N^2-isogeny Φ, because we can split Φ in two: $\Phi = \Phi_2 \circ \Phi_1$ with Φ_i a N-isogeny, and glue in the middle. We refer to [Rob23b; DLRW24] for more details.

We warn the reader that this strategy works only if the auxiliary isogeny α is given by a matrix of integers. That's because in that case we know the pushforward of α by ϕ Id (this is simply the same matrix because it commutes with ϕ Id), and we know the codomain of Φ already. Using a general auxiliary isogeny α in order to embed ϕ in Φ requires (in general) to know ϕ on the $N > n$ torsion.

Remark B.1. The group $E[N]$ is of cardinality N^2, so the condition $N^2 > n$ is not enough to uniquelty determine ϕ. By Section 3.3 we would need $N^2 > 4n$ instead. The reason for this discrepancy is as follows.

First, we describe Φ by its kernel, but that only determines Φ up to postcomposition by an automorphism of its codomain. When a principally polarised abelian variety (A, λ_A) splits as $A = A_1^{e_1} \times A_2^{e_2} \times \ldots$ as a product of principally polarised abelian varieties (A_i, λ_{A_i}) with λ_A the product polarisation of the λ_{A_i} and (A_i, λ_{A_i}) not isomorphic to (A_j, λ_{A_j}), then by the polarised Poincare's decomposition theorem, polarised automorphisms of (A, λ_A) are generated by the polarised automorphisms of the (A_i, λ_{A_i}) along with the permutations of the e_i factors of each A_i. In our setting, the codomain of Φ is a product of elliptic curve, whose automorphisms are ± 1 (unless we are in the special case of $j = 0, 1728$), so there is only a sign ambiguity for the individual isogenies composing Φ, hence in particular for ϕ. This is the same situation as if we only had the kernel of ϕ rather than ϕ.

Secondly, Φ embed several isogenies at once (in particular it embeds ϕ and its dual), and sometimes we require the extra information to identify ϕ among these isogenies. For instance, take a curve E with complex multiplication by $\sqrt{-2}$, and take the endomorphism $\phi = 3 + \sqrt{-2}$. Then ϕ and $-\widetilde{\phi} = -3 + \sqrt{-2}$ have norm 11 and act the same on the 6-torsion

(remark that $11 < 6^2 < 4 \times 11$). So while we can embed both of them at the same time into a higher dimensional 6-isogeny, to distinguish between them we need to evaluate them on more points.

We have seen in Section 6 ways to embed an isogeny $\phi : E_1 \to E_2$ into a higher dimensional isogeny without even knowing its codomain. For instance, if we have an efficient representation of an $n_1 n_2$-isogeny ϕ with $n_1 \wedge n_2 = 1$, we can split ϕ in $\phi = \phi_2 \circ \phi_1 = \phi_1' \circ \phi_2'$ and we can embed $\phi_1, \phi_2', \widetilde{\phi_1'}, \widetilde{\phi_2}$ at the same time into some higher dimensional isogeny Φ. One solution to identify which isogeny is which is to use pairings, see Lemma 6.2.

B.3. **The codomain product theta structure.** When we embed ϕ into an higher dimensional smooth N-isogeny Φ, and we compute Φ from its kernel using the theta isogeny algorithm described in Section 5.1.2, we have the following technical difficulty: by construction of the HD representation, the domain A and codomain B of Φ are a product of elliptic curves with their product polarisations. So we start with the product theta structure on A: we can efficiently convert between Weierstrass coordinates on E and level 2 or 4 theta coordinates on E (taking a small extension if needed to make the theta structure rational), and then we take the product theta structure to work on A: the resulting theta embedding on A is simply the Segre embedding.

However, when applying Lemma 5.7 to compute B, there is no reason that the resulting theta structure Θ_B on B is the product theta structure. Since we need to project back to elliptic curves to recover ϕ from Φ, as a first step we first need to convert Θ_B to a product theta structure Θ_B' (this is a linear change of variable). For this, we could simply test all the automorphisms of theta structure until we find a product theta structure. But, under the conditions of Lemma 5.7, we can actually predict which theta structure we will obtain on B, and know in advance which automorphism to apply to get a product theta structure in the end, see [DLRW24, Appendix F] for the technical details.

In dimension 2, the situation is simpler, because we can detect when we are on a product by the annulation of some even level 4 theta constant θ_i, and we are on a product theta structure precisely when $i = i_0 = (11; 11)$, and so it suffices to take any automorphism sending i to i_0 to recover a product theta structure (see [DMPR24, § 4]).

We note also that while Φ is completely determined by its kernel, to decompose Φ via the theta isogeny algorithm as in Lemma 5.7, we might need a bit more information. Typically, we embed ϕ into a 2^e-isogeny Φ in dimension g, and we work in level $m = 2$. To get a well defined theta structure on the codomain of Φ, we need points of 2^{e+2}-torsion above its kernel, and for that the most convenient way is to require to know how ϕ acts on the 2^{e+2}-torsion, not only the 2^e-torsion (of course we could guess it from the information we have).

B.4. **Gluing in the middle.** The same remark applies when we can split Φ in two: $\Phi = \Phi_2 \circ \Phi_1$, and we compute $\Phi_1 : A \to C$ and $\widetilde{\Phi_2} : B \to C$. For instance, if Φ is a 2^e-isogeny, we only need to know the action of ϕ on the $2^{e/2}$-torsion to recover the kernel of Φ_1 and $\widetilde{\Phi_2}$. But to be sure we recover the same theta structure on C from A and from B, we need to know (or guess) the action of ϕ on the $2^{e/2+2}$-torsion. We refer again to [DLRW24, Appendix F] and [Dar24, Appendices A, B] for more details. The advantage of imposing a theta structure of level m on C is that it kills the automorphisms of C (all automorphisms if $m \geq 3$, but if $m = 2$ there remain the ± 1 automorphisms), which solves the problem of gluing up to automorphisms we would have had if we had not rigidified C thusly.

Appendix C. Ideal to isogeny algorithms in the supersingular case and applications to the SQIsign family

In this section we give more details on the ideal to isogeny algorithms in the supersingular case. Among its many cryptographic applications, we can mention the SQIsign family of signature scheme, which we describe in Appendix C.6.

C.1. **Double paths.** In Section 4.2, we gave an overview of different algorithms to convert an ideal I to an isogeny $\phi_I : E_1 \to E_2$, assuming that we know an efficient representation of the endomorphism ring $\mathrm{End}(E_1)$ of E_1.

Actually, what we explained is only how to convert an equivalent ideal J into an isogeny $\phi_J : E_1 \to E_2$; where J is a smoothening of I in the first generations, and simply the smallest equivalent ideal in the fourth generation.

There remains two questions: how do we compute ϕ_I once we know how to compute ϕ_J? And how can we compute an efficient representation of $\mathrm{End}(E_2)$, which would be needed if we want to convert a new ideal I_2 from E_2?

For the first question, since we know I and J, we know the endomorphism $\alpha = \widehat{\phi_J} \circ \phi_I$; in terms of ideal this is a generator of the principal ideal $\bar{J}I$. Furthermore, by assumption we know how to evaluate it on E_1. And if $P \in E_1$, we have $\phi_J(\alpha(P)) = N(J)\phi_I(P)$. We can thus recover the image of P by ϕ_I up to the factor $N(J)$. This at least allow us to evaluate ϕ_I on all points P of torsion prime to $N(J)$. We have seen in Proposition 6.6 that we have a division algorithm which allows to evaluate ϕ_I efficiently when we know how to evaluate $N(J)\phi_I$ efficiently.

Another solution is to simply compute another "nice" ideal J' equivalent to I, of norm prime to $N(J)$. In other words: we build *a double path* of isogenies $\phi_J, \phi_{J'} : E_1 \to E_2$ where both isogenies have an efficient representation and are of coprime degrees. In fact, the Clapotis algorithm used in the fourth generation already directly constructs a double path. From this double path, we can translate any other ideal I into an isogeny ϕ_I efficiently, by evaluating a suitable endomorphism on E_1 first and then applying ϕ_J or $\phi_{J'}$.

The same solution works to evaluate endomorphisms on E_2. First we know that abstractly $\mathrm{End}(E_2)$ is the right order of I. Next, given $\beta \in \mathrm{End}(E_2)$, we can consider the endomorphism $\alpha = \widehat{\phi_J} \circ \beta \circ \phi_J$, which we know how to evaluate efficiently on E_1 by assumption. Since we know an efficient representation of ϕ_J, then by Proposition 6.4 we can build an efficient representation of $\widehat{\phi_J}$. But in fact, in our setting, when J is taken to be a smoothening of I, then it decomposes as a product of small degree isogenies so it is easy to compute its dual directly; and in the fourth generation where we use the Clapotis algorithm to embed a double path $\phi_{J_1}, \phi_{J_2} : E_1 \to E_2$ into a smooth HD isogeny Φ, then Φ is also decomposed into a product of small HD isogenies; hence we can also directly compute $\widehat{\Phi}$, and so $\widehat{\phi_{J_1}}, \widehat{\phi_{J_2}}$ too. In any case we can compute $\phi_J \circ \alpha \circ \widehat{\phi_J}(Q) = N(J)^2\beta(Q)$, hence recover $\beta(Q)$ at points of order prime to $N(J)$. The general case to evaluate β proceed by division or via a double path like for ϕ_I above.

In fact, it is often more practical to build everything from a special nice curve E_0, with particularly nice endomorphism evaluation (typically E_0 will be a supersingular curve defined over $\mathbb{F}_p$ and with small discriminant). Assume that we have built a isogenies $\phi_{J_1} : E_0 \to E_1$ and $\phi_{J_2} : E_0 \to E_2$ represented by ideals J_1, J_2. Then if $P \in E_1$, we have $N(J_1J_2)\phi_I(P) = \phi_{J_2} \circ \theta \circ \widehat{\phi_{J_1}}$, where $\theta = \widehat{\phi_{J_2}} \circ \phi_I \circ \phi_{J_1}$ is an endomorphism on E_0. This allows to evaluate any isogeny ϕ_I on points of order coprime to $N(J_1J_2)$. The same technique holds to evaluate endomorphisms on E_1 and E_2 at points of order coprime to $N(J_1)$ and $N(J_2)$ respectively. We can then invoke the division algorithm for the general case. Or, simpler, we only need

to compute a double "double path", i.e. an efficient double path $\phi_{J_1}, \phi_{J_1'}$ between E_0 and E_1, and another ϕ_{J_2}, ϕ_{J_2}' between E_0 and E_2 (with $N(J_1J_2)$ coprime to $N(J_1'J_2')$).

C.2. **The KLPT algorithm and Eichler orders.**

C.2.1. *KLPT.* The first generation of an ideal to isogeny algorithm is the KLPT *smoothening* algorithm from [KLPT14]. The idea is to find an ideal J equivalent to I, but of smooth norm.

If J is equivalent to I, ϕ_I, ϕ_J are both isogenies from E_1 to E_2, so $\widehat{\phi_J} \circ \phi_I$ is an endomorphism $\alpha \in \mathrm{End}(E_1)$, which is in I because the endomorphisms factorizes through ϕ_I by construction. It follows that $J = I\frac{\overline{\alpha}}{N(I)}$, which is of norm $N(\alpha)/N(I)$. So the goal is to find $\alpha \in I$ such that $N(\alpha)/N(I)$ is of smooth norm.

The KLPT algorithm shows that, if $\mathrm{End}(E_1)$ is a special quadratic order (called special extremal order), it is heuristically always possible to find (in polynomial time!) $\alpha \in I$ such that $N(\alpha)/N(I) = N$ for any N large enough: $N \gg p^3$. A proven algorithm (under GRH) was given in [Wes22], for $N = p^C$ but with no explicit bound on C. An example of a special extremal order is the elliptic curve $E_0 : y^2 = x^3 - x$ when $p \equiv 3 \mod 4$, whose endomorphism ring contains $\mathbb{Z}[i]$.

For the general case where E_1 is not special extremal, one can instead build a smooth connecting ideal J_1 between E_0 and E_1, and another J_2 between E_0 and E_2. Then $J = J_2\overline{J_1}$ is a smooth connecting ideal between E_1 and E_2. But we can only expect to find such a J of norm $N \gg p^6$, because $N(J_i) \gg p^3$.

Taking N powersmooth, this gives an algorithm which is efficient in theory, but not in practice (but see [EPSV] for several tricks). Indeed, the size of N means that we cannot expect to find rational N-torsion, we need to take field extensions.

C.2.2. *Eichler orders.* The first practical algorithm to convert an arbitrary ideal to an isogeny was given in [DKLPW20]: using the theory of Eichler order, it is proven that one can (heuristically) always find a smooth ideal J connecting E_1 and E_2 of norm $N \gg p^{4.5}$. More precisely, let I_0 be an ideal connecting E_0 to E_1. Then one can (heuristically) find an ideal J equivalent to I and of norm N as long as $N \gg p^{1.5}N(I)^3N(I_0)^3$. By Minkowski's bound, the (reduced) norm of the smallest ideal connecting two supersingular curves is in $O(\sqrt{p})$. Replacing I and I_0 by small equivalent ideals, we obtain the $p^{4.5}$ bound mentioned above. And the closer E_1 is to E_0 (compared to a uniform supersingular curve), the better the bound, up to p^3 when $E_1 = E_0$.

Let us quickly explain how Eichler orders are used. Let E_0 be a special extremal curve, and I_0 the ideal connecting E_0 to E_1, and assume that it is coprime to I for simplicity. Let $I' = I_0^*I$ be the pullback ideal of I (in term of isogeny this corresponds to the isogeny $\phi_I : E_1 \to E_2$ pullbacked to $\phi_{I'} : E_0 \to E_0'$ via $\phi_{I_0} : E_0 \to E_1$). Since E_0 is special extremal, by KLPT we can find a nice equivalent ideal J' to I' of norm N for $N \approx p^3$. We thus have two isogenies: $\phi_{I'}, \phi_{J'} : E_0 \to E'$. But while $\phi_I : E_1 \to E_2$ is the pushforward of $\phi_{I'}$ to E_1, the pushforward ϕ_J of $\phi_{J'}$ to E_1 may give an isogeny $E_1 \to E_2'$ with a different codomain than E_2!

Let $K = E_0[I_0]$ be the kernel of $\phi_{I_0} : E_0 \to E_1$. We have an endomorphism $\alpha = \widehat{\phi_{J'}} \circ \phi_{I'}$, and in term of ideals $J' = I'\overline{\alpha}/N(I')$. One can show that the codomain of ϕ_J is still the same E_2 if and only if $\alpha(K) = K$ [Ler22b, Proposition 2.3.11]. Thus, for the above construction to work, J' cannot be taken to be any ideal equivalent to I', it has to come from some α such that $\alpha(K) = K$.

The endomorphisms of E_0 such that $\alpha(K) = K$ form precisely the Eichler order $O = O_0 \cap O_1$ where $O_0 = \mathrm{End}(E_0)$ and $O_1 = \mathrm{End}(E_1)$. We refer to [Ler22b; Arp22] for more details on Eichler order and their relationship with kernel of isogenies.

Thus the KLPT algorithm has to be adapted to sample elements α that are in the Eichler order O, in order for the pushfoward strategy to work. We refer to [Ler22b; DKLPW20] for these algorithms, which give the bound described above.

C.3. **Ideal to isogeny: splitting a large dimension one 2^e-isogeny.** For practical instanciation of KLPT, we would like to:

(1) Use smooth ideals with a very small smoothness bound, ideally take $N = 2^e$
(2) Work over $\mathbb{F}_{p^2}$ and avoid taking any field extension.

Unfortunately, this cannot happen because, even with a carefully chosen p, $E(\mathbb{F}_{p^2}) = (\mathbb{Z}/(p \pm 1)\mathbb{Z})^2$ can fit at most the 2^f torsion with $2^f \mid p \pm 1$. While the KLPT algorithm (the improved version with Eichler orders) can only generate 2^e-isogenies with $2^e \approx p^{4.5}$.

C.3.1. *Using E_0 to refresh the torsion.* As mentioned in Section 4.2, the first SQIsign paper [DKLPW20] introduced the idea of splitting the large 2^e-isogeny into chunks of 2^f-isogenies. First, select a p with large available rational (over $\mathbb{F}_{p^2}$) 2^f-torsion, and search for J equivalent to I of norm 2^e, $2^e \approx p^{15/4}$ (this is better than the bound of $p^{4.5}$ above, because in the context of SQIsign E_1 is only at distance $p^{1/4}$ from E_0 and $p^3 p^{3/4} = p^{15/4} = p^{3.75}$).

If we split the isogeny ϕ_J into chunks of 2^f-isogenies, then $J_1 = J + (2^f)$ is the ideal corresponding to the first chunk: $\phi_{J_1} : E_1 \to E_{1,1}$. Let $J' = J_{1,*}J$ the pushforward of J by J_1, we now need to split J' into chunks to evaluate the next step $\phi_{J_2} : E_{1,1} \to E_{1,2}$ from $E_{1,1}$, where $J_2 = J' + (2^f)$. The problem is that to compute the next kernel, $E_{1,1}[J_2] = E_{1,1}[2^f, J']$, we need to evaluate J' on the 2^f-torsion, and we cannot go back to E_1 because J_1 is of reduced norm 2^f, so we have "consumed" all our available torsion when we evaluated ϕ_{J_1}.

We need to "refresh" the 2^f-torsion on $E_{1,1}$. The solution in SQIsign is to build a T-isogeny $\phi_T : E_0 \to E_{1,1}$ with T smooth and odd. We can then use this isogeny from E_0 to evaluate the action of J' on the 2^f-torsion of $E_{1,1}$, since it is of degree coprime to 2^f. Since E_0 is special extremal we can build an isogeny ϕ_T as long as $T \gg p^3$. This is still bigger than the available torsion, but we are saved by two things.

The first one, is that in that case we already know the codomain $E_{1,1}$, we don't need to construct it. This means that we can split ϕ_T in two: write $T = T_1 T_2$, ϕ_T as $\phi_{T_2} \circ \phi_{T_1} : E_0 \to E' \to E_{1,1}$ for some random intermediate curve E'. We can compute ϕ_{T_1} from its kernel in $E_0[T_1]$, and $\widetilde{\phi_{T_2}}$ from its kernel in $E_{1,1}[T_2]$. In that case, since T_2 is odd we can use ϕ_{J_1} to go back to E_1 to evaluate suitable endomorphisms on $E_{1,1}[T_2]$ to obtain $\operatorname{Ker} \widetilde{\phi_{T_2}}$ (in practice, by also using a suitable isogeny connecting E_0 to E_1). We then just need to glue things in the middle at E'. This trick of splitting an isogeny in two, when both the domain and codomain are already know is very useful, and allows to get a square factor in the amount of effective available torsion (see Appendix B for the same trick for the HD representation).

So we can expect T_1, T_2 to be of size roughly $p^{3/2}$, which is still bigger than p. The next trick is to work both with the curves $E_0, E_{1,1}$ and their quadratic twists. This allows to have both the $p-1$-torsion and the $p+1$-torsion available, while still working over $\mathbb{F}_{p^2}$ (the only condition is to not use them at the same time, which would involve working over the quadratic extension $\mathbb{F}_{p^4}$).

Example C.1. In the original SQIsign, one use available 2^f-torsion with $2^f \approx p^{1/8}$ (so that the response is split into ≈ 30 blocks of 2^f-isogenies), and T_1^2-torsion (i.e., $T_2 = T_1$) with $T_1 \approx p^{3/2}$, split into one part dividing $p-1$ and the other part dividing $p+1$. There remain

approximatively $\approx p^{1/4}$ available torsion not used, because it is very hard to find such smooth primes with smoothness conditions both on $p-1$ and $p+1$ already, see [BSC+23].

For a security parameter λ, p has size 2λ, and the signature has size around $23/2 \cdot \lambda$.

C.3.2. *Using an endomorphism to refresh the torsion.* In [DLLW23], a novel idea is to use an endomorphism γ of reduced norm T on $E_{1,1}$ instead to refresh the 2^f-torsion. This endomorphism γ (which need to satisfy some technical conditions) can be evaluated from the 2^f-isogeny $\phi_{J_1} : E_1 \to E_{1,1}$ because T is odd. The advantage compared to building a T-isogeny $E_0 \to E_{1,1}$ is that one can find a suitable endomorphism γ of smooth norm T for $T \approx p^{2.5}$, compared to p^3 for the isogeny; splitting it in two we only need accessible torsion of size $\approx p^{1.25}$ compared to $\approx p^{1.5}$. This leaves more room for the rest of the torsion, makes it easier to find good primes p, and allows to increase the available 2^f-torsion.

We briefly explain how the algorithm works. The idea is that we know the kernel K_1 of the contragredient isogeny $E_{1,1} \to E_1$, it is given by $\phi_{J_1}(E_1[2^f])$. What we want is to find the kernel $K = E_{1,1}[J_2]$ of the next isogeny $E_{1,1} \to E_{1,2}$. If we can find an effective representation of any endomorphism $\gamma \in \mathrm{End}(E_{1,1})$ such that $\gamma(K_1) \cap K_1 = 0$ (this condition can be translated into the fact that γ should not be in some Eichler order), then by linear algebra we can find an endomorphism $\gamma' = a\gamma + b$ such that $\gamma'(K_1) = K$. Evaluating γ on K_1 thus allow us to find our next kernel K.

Example C.2. For SQIsign, using an endomorphism rather than an isogeny from E_0 drops the requirement on the smooth accessible torsion T from $T \approx p^{3/2}$ to $T \approx p^{5/4}$. The relaxed torsion requirement makes finding suitable p easier and improves the signing time by enabling the use of primes p with $2^f \approx p^{1/4}$ accessible torsion, which increases the size of each block of 2^f-isogenies and reduces the number of steps to ≈ 15. Furthermore, the signature is compressed further and has size around $17/2 \cdot \lambda$. See Appendix C.6.1 for more details.

C.3.3. *Using an endomorphism represented by a dimension* 2 *isogeny.* In [Ler23b], using tools from [CLP24], Leroux introduces the first version of an ideal to isogeny algorithm using the newer HD representation. Namely, rather than searching for γ of smooth norm in order to be able to evaluate it efficiently, he uses a dimension two 2^f-representation of γ. As we have seen in Section 5, an HD representation allows to efficiently represent any γ of odd norm $T < 2^f$ (even non smooth), as long as we know how γ acts on the 2^f torsion. In fact, by the same splitting trick as above, we can work up to γ of norm $T < 2^{2f}$. Since we need to be able to evaluate γ on the 2^f-torsion to obtain this representation, the only change is that J is taken to be a 3^h-isogeny rather than a 2^e-isogeny as previously, and it is split into chunks of 3^g-isogenies rather than chunks of 2^f-isogeny. For instance, one can take $p = 2^f 3^g u - 1$ for a small cofactor u. Since we relax the smoothness bound on γ, one can find a suitable endomorphism γ of reduced norm $T \approx p^{2/3}$. We thus only require $2^f \approx p^{1/3}$ to represent γ, leaving ample room for the chunks of 3^g-isogenies.

Rather than computing chunks of 3^g-isogenies in dimension 1, and using T-endomorphisms embedded in 2^f-isogenies in dimension 2, we could reverse the role of 2 and 3, but going from 2-isogenies to 3-isogenies is less expensive in dimension 1 than dimension 2.

C.3.4. *Using an HD representation for isogenies from* E_0. A recent alternative, in [ON24], is to go back to the original SQIsign protocol, using isogenies from E_0 rather than endomorphisms, but like in [Ler23b] to use an HD representation to relax the smoothness requirement on these isogenies, as was pioneered in SQIsignHD [DLRW24]. In that version, one take $p = 2^f u - 1$

for a small cofactor u, so that the full $2^f \approx p$ is available, and we write $f = f_1 + f_2$. The isogeny ϕ_I is split into chunks of 2^{f_1}-isogenies, and the refreshing isogenies $\phi_T : E_0 \to E_{1,1}$ of degree T are represented by an HD 2^{f_2}-representation, which requires to be able to evaluate ϕ_T on the 2^{f_2}-torsion.

Using a previously built isogeny $\phi'_T : E_0 \to E_1$, and the isogeny $\phi_{J_1} : E_1 \to E_{1,1}$, we can indeed recover the action of ϕ_T on the 2^{f_2}-torsion: ϕ_{J_1} is a 2^{f_1}-isogeny, so only consumes 2^{f_1} out of our available $2^{f_1+f_2}$-torsion!

One can use a dimension 4 HD representation, like in SQIsignHD. This allows to split ϕ_T in two and allows to use any T such that $T < 2^{2f_2}$. Or we can use the fact that E_0 contains $\mathbb{Z}[i]$ to build a dimension two representation, which will be more efficient that the dimension four representation (see Remark 5.17); but we cannot split in two in this case, so we need $T < 2^{f_2}$. We already mentioned that Minkowski's bound show that we can always find an isogeny $\phi_T : E_0 \to E_{1,1}$ of (non smooth) norm T for $T = O(\sqrt{p})$. So even the dimension 2 representation leaves ample room to split ϕ_I into chunks of 2^{f_1}-isogenies.

C.4. **The Clapotis algorithm.** Another recent alternative is given in [BDD+24], inspired by the Clapoti(s) algorithm from [PR23b], see Example 6.13. It allows to directly convert any isogeny $\phi_I : E_1 \to E_2$ in one go by using a dimension 4 HD-representation embedding at the same time two isogenies $\phi_{J_1}, \phi_{J_2} : E_1 \to E_2$, with $N(J_1)$ coprime to $N(J_2)$. In other words this algorithm builds a double path from E_1 to E_2 in one go by using a dimension 4 2^e-isogeny. The constraint on torsion is $2^e \approx N(J_1)N(J_2)$, and taking small equivalent ideals $J_1, J_2 \sim I$ we can have $N(J_1)N(J_2) \approx p$, which is barely enough to fit into the available 2^f-torsion (we cannot split in two in that case because we don't know E_2 yet). Then ϕ_I can be recovered from this double path and the effective endomorphism representation $\mathrm{End}(E_1)$ on E_1.

In practice, it is better, as explained in [BDD+24], to build a double path between E_0 and E_1, and another between E_0 and E_2. Because $\mathrm{End}(E_0)$ contains $\mathbb{Z}[i]$, which gives a lot of small endomorphisms, this allows to find a dimension 2 representation of the double paths (again, see Remark 5.17). This replaces the need to compute one dimension 4 isogeny by two dimension 2 isogeny, which is much faster in practice.

Of course, the parameters given in the discussions above, like splitting a big 2^e-isogeny into chunks of 2^f-isogeny, or using HD representations to embed T-isogenies into higher dimensional 2^f-isogeny could be generalised to any $B \mid B'$ with B, B' smooth (in our examples, $B = 2^f$ and $B' = 2^e$). This replaces the constraint of having accessible 2^f-torsion to having accessible B-torsion. But 2^e-isogenies are the one we prefer to use in practice, because they are the most convenient, especially in higher dimension. When we have the choice of the prime p, so when building protocols, we might as well choose p to have large accessible 2^f-torsion.

C.5. **Improving KLPT?.** From the diameter of the supersingular ℓ-isogeny graph, we know that there exist isogenies of norm $\ell^e \approx p$ between any two elliptic curves E_1, E_2. But currently the KLPT algorithm, even using the improved variant via Eichler orders, can only generate isogenies of norm $\ell^e \approx p^{4.5}$ for generic elliptic curves (for SQIsign the parameters allow to reach $\ell^e \approx p^{15/4}$). This is too big to fit into the available 2^f-rational torsion, even taking $p = u2^f - 1$ a pseudo-Mersenne prime. So for the SQIsign signature, this large 2^e-isogeny has to be split into chunks of 2^f-isogenies as we have seen. And even for the verification,

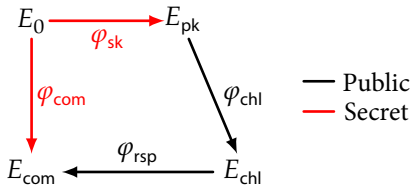
taking f as large as possible like in AprèsSQI [CEMR24] (which impacts the signature time), the verifier still has to compute several 2^f-isogenies in dimension 1 with $2^f \approx p$.

By contrast, the Clapotis algorithm can directly convert the smallest ideal between E_1 and E_2, which is of norm $\approx \sqrt{p}$ (or less) by Minkowski's theorem, and this ideal can be efficiently found by the LLL algorithm. Hence the ideal can be converted into an isogeny in one step, which drastically speeds up the signature; and the verification only needs a 2^e-isogeny in dimension 2 with $2^e \approx \sqrt{p}$. From the current state of the art, it seems that computing this smaller $2^e \approx \sqrt{p}$-isogeny for verification in dimension 2 is faster than computing a degree $2^e \approx p^{15/4}$ isogeny in dimension 1. Still, [CEMR24] gives several tricks using uncompressed signatures and hints to speed up verification in dimension 1, so SQIsign might still stay competitive with SQIsign2d for verification, thanks to AprèsSQI (but in any case the signature will be much faster in SQIsign2d, see Appendix C.6).

However, being able to generate directly an equivalent smooth ideal J of norm $2^e \approx p$ would be enough to fit into the available 2^f-torsion, and allow to convert it into an isogeny in one step, as is also currently done in the fourth generation, all the while staying in dimension 1 rather than going in dimension 2. The verification would also be faster: since we expect a factor $\approx \times 4$ (at least) slowdown between dimension 1 and 2 as explained in Section 5.1, we expect an isogeny of dimension 1 of norm $\approx p$ to be at least twice as fast to compute as a $\approx \sqrt{p}$-isogeny in dimension 2.

C.6. **The SQIsign family.** We conclude this section by comparing different versions of SQIsign (which use different generations of an ideal to isogeny algorithm), along with their concrete instantiation for NIST level 1 (which means that the security parameter is $\lambda = 128$ bits). In all cases, for a security parameter λ, we work with supersingular curves over $\mathbb{F}_{p^2}$ with p of size 2λ.

We recall the general structure of the SQIsign identification protocol: the Prover wants to prove the knowledge of a secret isogeny $\varphi_{\mathsf{sk}} : E_0 \to E_{\mathsf{pk}}$; where only the codomain E_{pk} is public. He computes a secret commitment isogeny $\varphi_{\mathsf{com}} : E_0 \to E_{\mathsf{com}}$, and publishes E_{com}. The verifier challenges by an isogeny $\varphi_{\mathsf{chl}} : E_{\mathsf{pk}} \to E_{\mathsf{chl}}$. The prover responds by an isogeny $\varphi_{\mathsf{rsp}} : E_{\mathsf{chl}} \to E_{\mathsf{com}}$ connecting the challenge curve to the commitment curve. See for instance [BDD+24] for more details.



C.6.1. *SQIsign.* This is the version in [DKLPW20] with the improvements of [DLLW23]. This version uses dimension 1 both for the signature and the verification, and the signature takes $\approx 17/2 \cdot \lambda$ bits. The verification is reasonably fast but the signature is very slow, and is very hard to implement in constant time due to the many operations on the quaternion side required. Furthermore the security assumptions rely on many ad-hoc heuristics.

The public key is given by an isogeny of degree $\approx p^{1/4}$, which is not enough to reach the whole supersingular keyspace (this would require a degree $\approx p^{1/2}$). In particular, it is not statically uniform (a provable bound to reach the uniform distribution up to $2^{-\lambda}$ would even require a degree $\approx p^2$). The commitment is a smooth isogeny of degree $\approx p$. The response

is an isogeny of degree $2^e \approx p^{3.75}$ which is split into 15 isogenies of degree $2^f \approx \sqrt{p}$. The torsion is refreshed using 15 endomorphisms of smooth degree $T \approx p^{5/4}$.

For the NIST submission, at level 1, we have p with accessible 2^f-torsion and T-torsion with $f = 75$ and $T = 3^{36} \cdot 7^4 \cdot 11 \cdot 13 \cdot 23^2 \cdot 37 \cdot 59^2 \cdot 89 \cdot 97 \cdot 101^2 \cdot 107 \cdot 109^2 \cdot 131 \cdot 137 \cdot 197^2 \cdot 223 \cdot 239 \cdot 383 \cdot 389 \cdot 491^2 \cdot 499 \cdot 607 \cdot 743^2 \cdot 1033 \cdot 1049 \cdot 1193 \cdot 1913^2 \cdot 1973$. The commitment is of degree $T/3^g$ ($g = 36$), and the challenge is of degree $2^f 3^g$. The available 2^f torsion is enough to split the response in 14 rather than 15, so the signature needs to compute 14 2^f-isogenies and 13 endomorphisms of degree T. The verification needs to compute the challenge again along with the 14 2^f-isogenies in dimension 1. The signature takes 177 Bytes.

C.6.2. *SQIsignHD.* This is the version in [DLRW24], which actually describes two variants.

FastSQIsignHD: this version uses dimension 1 for the signature and dimension 4 for the verification, and the signature takes $\approx 13/2 \cdot \lambda$ bits. The prime p is a SIDH prime of the form $p = 2^f 3^g - 1$, with $2^f \approx 3^g \approx \sqrt{p}$. The public key and commitment are generated by a double path procedure, which generates a double path of degree 2^{2f} and $3^{2g} \approx p$ to the same codomain. In practice, the double path is computed via three 2^f-isogenies and three 3^g-isogenies in dimension 1. This double path procedure is assumed to be computationally uniform. The challenge is generated by an isogeny of degree 3^g. The response is an isogeny of degree $q \leq 2^{2e}$ such that $2^{2e} - q$ is a sum of two squares, this allows to find a dimension 4 HD representation of this response. In practice, we take $2e \approx \log_2(p) + 16$.

The signature only needs to compute the commitment, the challenge is represented by its kernel, and the response by its action on the 2^f-torsion. The verification computes the challenge from its kernel, and compute the response via its dimension 4 representation, namely it needs to compute two 2^e-isogenies in dimension 4 with $e \approx \log_2(p)/2 + 8 \leq f$.

For NIST level 1, we have $p = 13 \cdot 2^{126} 3^{78} - 1$, and the signature takes 109 Bytes. The commitment needs three 2^{126}-isogenies and three 3^{78}-isogenies in dimension 1.The verification computes the challenge of degree 3^{78} and checks the response via two 2^{73}-isogenies in dimension 4.

RigorousSQIsignHD: this version uses dimension 1 for the signature and dimension 8 for verification. This version is optimised for security: the public key, and commitment are provably statically uniform, and the zero knowledge property has a clean proof which relies on a RADIO oracle. However, while this protocol is polynomial time with respect to the security parameter λ, it is much too slow to be used in practice and no serious instantiation has been proposed.

Indeed, the public key, challenge and commitments are given by isogenies of degree $\approx p^3$, this requires accessible T-torsion with $T \approx p^3$, hence imposes to work with field extensions. The response is an isogeny of degree $q \approx p^2 \leq 2^{2e}$, which will be computed during the verification via two 2^e-isogenies in dimension 8, with $2^e \approx p$. This imposes to take $p = c2^e - 1$, and T to be powersmooth. The signature thus needs to split large degree isogenies into block of T-isogenies with the T-torsion defined over extensions, so is much slower than in SQIsign. The verification needs to compute two $2^e \approx p$-isogenies in dimension 8, compared to FastSQIsignHD which only needs two $2^e \approx \sqrt{p}$-isogenies in dimension 4. A 2^n-isogeny in dimension 8 is expected to be roughly 32 times slower than a 2^n-isogeny in dimension 4, so the verification time of RigorousSQIsignHD would also be much slower than in FastSQIsignHD.

C.6.3. *SQIsign2d.* . In this section, we will focus on the variant SQIsign2d-West of [BDD+24], but see also [NO24; DF24]. This version has faster signature and verification times than SQIsign, a shorter signature and a cleaner security proof.

The prime p is chosen to be a Montgomery friendly prime of the form $p = c2^e - 1$ so that the 2^e-torsion is accessible, with $2^e \approx p$. The signature takes $\approx 8 \cdot \lambda$ bits. The public key is provably statically uniform and computed via the Clapotis version of the ideal to isogeny algorithm applied a uniformly random ideal class. The Clapotis algorithm requires one 2^e-isogeny in dimension 2, and between zero and two 2^f-isogenies in dimension 2, with and $2^f \approx \sqrt{p}$, depending on the chosen trade off between work on the quaternion side and on the isogeny side. The commitment is also statically uniform and use the same algorithm. Like in RigorousSQIsignHD, there is a clean zero knowledge proof which relies on a slightly different oracle.

Like in FastSQIsignHD the signature only generates the kernel of the challenge, which is computed during the verification. This challenges is a 2^e-isogeny in dimension 1. The verification also checks the response, via a 2^f-isogeny in dimension 2. Compared to FastSQIsignHD, the signature step also needs to compute an auxiliary isogeny $\beta : E_{\mathsf{chl}} \to E_{\mathsf{aux}}$ of appropriate degree to embed the response into a 2^f-isogeny $\Phi : E_{\mathsf{chl}} \times E_{\mathsf{aux}} \to E_{\mathsf{com}} \times E'_{\mathsf{aux}}$ in dimension 2, this is also done via the Clapotis version of the ideal to isogeny algorithm. More precisely, it is $\widetilde{\Phi}$ which is computed first from the commitment curve and $\beta' : E_{\mathsf{com}} \to E'_{aux}$, and then $\widetilde{\Phi}$ is evaluated to recover Φ. The reason we want Φ rather than $\widetilde{\Phi}$ to embed the response is to compress the signature by using commitment recoverability. See [BDD+24, Figure 2] for the full diagram.

There is a slightly more heuristic version where the commitment is computed using the QFESTA algorithm (Example 6.12) rather than Clapotis (Example 6.13), which is faster but lack the provable statical uniform property. The challenge is also taken to be of degree 2^f, this allows to compute β hence Φ directly from E_{chl}.

For NIST level 1, we have $p = 5 \cdot 2^{248} - 1$ and the signature takes 148 Bytes using hints to speed up the verification. Removing hints could reduce the signature size up to 128 Bytes, at the cost of a slight increase (5 to 10 percent) to the verification time.

The signature computes the commitment via Clapotis, which uses a 2^{248}-isogeny and two 2^{126}-isogenies in dimension two. The auxiliary isogeny β' is also generated by Clapotis, this gives a 2^{126}-isogeny $\widetilde{\Phi}$ in dimension 2, and computing it allows to recover Φ. The verification computes the challenge, which is a 2^{248}-isogeny in dimension one, and the response is checked via a 2^{126}-isogeny in dimension two.

In the heuristic version the commitment only needs one 2^{248}-isogeny in dimension 2. The challenge is a 2^{122}-isogeny in dimension 1 which is computed during the verification. This allows to build Φ directly from β, computed via Clapotis. This heuristic version gives a signature roughly 60 percent faster than the cleaner version. The response is checked via a 2^{126}-isogeny in dimension two too, so the verification time does not change much (only the challenge is smaller).

The verification time of SQIsign2d is the fastest out of all the variants described in this section, and the signature is only slightly slower and less compact than FastSQIsignHD and much faster than SQIsign. But as explained in [BDD+24, Remark 24], we can adapt FastSQIsignHD to the prime used in SQIsign2d, this gives an even faster signature time than the original FastSQIsignHD. That version has a signature time roughly 3× faster than the heuristic version of SQIsign2d. However, this comes at the cost of the verification time, which needs to compute an isogeny in dimension 4 rather than 2, for an expected slow down of roughly ×8.

Appendix D. The class group of a non maximal order

Let R be an order in some number field K, R_0 be the maximal order of K, and $\mathfrak{f}$ the conductor ideal of R: $\mathfrak{f} = (R : R_0)$. This is both an ideal in R and in R_0.

The conductor square

$$\begin{array}{ccc} R & \longrightarrow & R_0 \\ \downarrow & & \downarrow \\ R/\mathfrak{f} & \longrightarrow & R_0/\mathfrak{f} \end{array}$$

is a Milnor square (also called an excision datum): it is both a pullback and a pushforward.

In particular, $\operatorname{Spec} R = \operatorname{Spec} R_0 \coprod_{\operatorname{Spec} R_0/\mathfrak{f}} \operatorname{Spec} R/\mathfrak{f}$ is given by the gluing of $\operatorname{Spec} R_0$ and $\operatorname{Spec} R/\mathfrak{f}$ over $\operatorname{Spec} R_0/\mathfrak{f}$. This is a pushout in the category of schemes, and gives an explicit description of $\operatorname{Spec} R$ as a blow down of the regular scheme $\operatorname{Spec} R_0$.

As shown by Milnor, finitely presented vector bundles (so in particular line bundles) satisfy excision, which means that that to specify a line bundle $\mathcal{L}$ on $\operatorname{Spec} R$ is the same thing as specifying a line bundle $\mathcal{L}_0$ on $\operatorname{Spec} R_0$ and $\mathcal{L}_\mathfrak{f}$ on $\operatorname{Spec} R/\mathfrak{f}$, along with an isomorphism $\mathcal{L}_0 \simeq \mathcal{L}_\mathfrak{f}$ over $\operatorname{Spec} R_0/\mathfrak{f}$. For vast generalisations of this result, see [BM21; EHIK21; AHHR24].

From this we obtain a Mayer-Vietoris exact sequence:

$$1 \to U(R) \to U(R/\mathfrak{f}) \oplus U(R_0) \to U(R_0/\mathfrak{f}) \to \operatorname{Pic}(R) \to \operatorname{Pic}(R_0) \oplus \operatorname{Pic}(R_0/\mathfrak{f}) \to 0,$$

which gives the usual exact sequence between $\operatorname{Pic}(R)$ and $\operatorname{Cl}(R_0)$:

$$1 \to R_0^*/R^* \to (R_0/\mathfrak{f})^*/(R/\mathfrak{f})^* \to \operatorname{Pic}(R) \to \operatorname{Cl}(R_0) \to 0. \tag{4}$$

This geometric derivation of Equation (4) might seem overkill, but it has the following nice application for isogenies between elliptic curves. Let $\phi : E_1 \to E_2$ be an horizontal isogeny between R-oriented elliptic curves, so that $\phi = \phi_I$ for some invertible ideal $I \in \operatorname{Pic}(R)$. The conductor ideal $\mathfrak{f}$ of $R \subset R_0$ gives ascending isogenies $\phi_{\mathfrak{f},1} : E_1 \to E_1'$ and $\phi_{\mathfrak{f},2} : E_2 \to E_2'$. Looking at the image $I' = IR_0 \in \operatorname{Cl}(R_0)$ of I under Equation (4), we also have an horizontal isogeny $\phi_{I'} : E_1' \to E_2'$, which commutes with ϕ_I under the ascending isogenies (and in fact is a pushforward: $\operatorname{Ker} \phi_{\mathfrak{f},2} = \phi_I(\operatorname{Ker} \phi_{\mathfrak{f},1})$).

Conversely, take some ideal class $[I'] \in \operatorname{Cl}(R_0)$ represented by I' giving some isogeny $\phi_{I'} : E_1' \to E_2'$. Let K_1 be the kernel of the descending isogeny $\widetilde{\phi_{\mathfrak{f},1}} : E_1' \to E_1$. If we take I' to be of prime norm to the conductor of R, the image $K_2 = \phi_{I'}(K_1)$ of K_1 by $\phi_{I'}$ is the kernel of a descending isogeny $\widetilde{\phi_{\mathfrak{f},2}} : E_2' \to E_2$, and we obtain an isogeny $\phi_I : E_1 \to E_2$ completing the pushforward square. Changing the class of I' amount to postcomposing by an endomorphism $\alpha : E_2' \to E_2'$; in general we obtain a different isogeny $E_1 \to E_2$ unless $\alpha(K_2) = K_2$. We can thus reinterpret isogenies $E_1 \to E_2$ as isogenies $E_1' \to E_2'$ with some extra level structure information (namely the image of the kernel K_1). In other words: E_2' is determined from E_2 and K_2, and it is also determined by the class of I. From the conductor square, we know that the class of I is determined by the class of some invertible ideal I' in R_0, along with some invertible ideal in $R/\mathfrak{f}$ and some gluing data. But we know that E_2 is given by the class of I', so this means that the data of K_2 should correspond to some module over $R/\mathfrak{f}$. One can make this statement rigourous by using the fact that the usual ideal to isogeny functor extends to modules, see Remark 4.1.

More generally, there are two ways to extend the usual relationship between ideals and isogenies to include level structure information. The first is to consider ideals with respect to a suborder whose conductor is related to the level structure. For instance, extending the Deuring's correspondence to keep track of the kernel of a specific isogeny $E_0 \to E_1$

corresponds to looking at ideals in the Eichler order $O_0 \cap O_1$. This is the point of view adapted in [Ler22b; Arp22].

Another approach is to use again that the ideal to isogeny functor extend to modules. For instance, if $R = \mathrm{End}(E)$, the map associated to the module $R \hookrightarrow R/I$ is the inclusion $E[I] \subset E$; in particular $E[n]$ corresponds to the module R/nR. Keeping track of level structure along isogenies corresponds to looking at modules over R along with modules over the endomorphism data of the level structure, with some compatible gluing conditions. Again, the conductor square allows to go back and forth between the two approaches (for sufficiently nice modules).

In other words, as argued in [PR23a], the module point of view, which extends the usual relationship between ideals and isogenies (as described in Section 4), is very fruitful since it can handle in a unified framework level structures and the higher dimensional isogeny graph starting from E^g. And the geometric description of $\mathrm{Spec}\, R$ as an excision datum of $\mathrm{Spec}\, R_0$ and $\mathrm{Spec}\, R/\mathfrak{f}$ over $\mathrm{Spec}\, R_0/\mathfrak{f}$ allows to better understand modules on R.

Appendix E. The Kodaira-Spencer isomorphism for abelian varieties

In this section, we briefly recall the Kodaira-Spencer isomorphism. IF A/k is an abelian variety, the Kodaira-Spencer map is a canonial isomorphism between $\mathrm{Sym}^2(T_0(A))$ and the tangent $T_A\mathcal{A}_g$ to A on the moduli space of principally polarised abelian varieties $\mathcal{A}_g$. In other words, deformations of A are controlled by the Sym^2 of differentials on A.

Let $p\colon A \to S$ be an abelian scheme over a smooth base. Recall that we have a canonical flat connection on the De Rham (hyper)cohomology, the Gauss-Manin connection:

$$\nabla\colon R^1p_*\Omega_{A/S} \to R^1p_*\Omega_{A/S} \otimes \Omega^1_S.$$

Combining the Gauss-Manin connection with the Hodge filtration, one can define the *Kodaira–Spencer map* (see [And17, § 1.4; FC90, § III.9]):

$$\kappa\colon T_S \to R^1p_*T_{A/S},$$

where $T_{A/S}$ denotes the dual of $\Omega^1_{A/S}$. Since $\mathrm{Lie}_S A = p_*T_{A/S} = s^*T_{A_S}$ where $s\colon S \to A$ is the zero section [EGM12, Prop. 3.15], by the projection formula [Stacks, Tag 0943], we have

$$R^1p_*T_{A/S} = \mathrm{Lie}_S(A) \otimes_{O_S} R^1p_*O_A.$$

Moreover, $R^1p_*O_A$ is naturally isomorphic to $\mathrm{Lie}_S(A^\vee)$, where $A^\vee \to S$ denotes the dual of A. Therefore, we can also write the Kodaira–Spencer map as

$$\kappa\colon T_S \to R^1p_*T_{A/S} \simeq \mathrm{Lie}_S(A) \otimes_{O_S} \mathrm{Lie}_S(A^\vee).$$

The Kodaira-Spencer map κ is invariant by duality. A polarization $A \to A^\vee$ induces another version of the Kodaira–Spencer map:

$$\kappa\colon T_S \to \mathrm{Sym}^2 \mathrm{Lie}_S(A) = \mathrm{Hom}_{\mathrm{Sym}}(\Omega^1_{A/S}, \Omega^{1\vee}_{A^\vee/S}) = \mathrm{Hom}_{\mathrm{Sym}}(\mathrm{Lie}_S(A)^\vee, \mathrm{Lie}_S(A^\vee)).$$

If we apply this construction to the universal abelian scheme $\mathcal{X}_g \to \mathcal{A}_g$, the Kodaira–Spencer map is an isomorphism [And17, § 2.1.1]. In particular, if $x\colon \mathrm{Spec}\, k \to \mathcal{A}_g$ is a point represented by a principally polarised abelian variety A/k, we have a canonical isomorphism $T_x\mathcal{A}_g \simeq \mathrm{Sym}^2(T_0(A))$. Moreover, if j is a modular invariant (i.e. a rational map $\mathcal{A}_g \to \mathbb{A}^1$), then via the Kodaira–Spencer isomorphism, its differential dj becomes a Siegel modular function of weight Sym^2.

Example E.1. In [KPR24], we use the Kodaire-Spencer isomorphism to work with *deformation representations* of isogenies in dimension 2.

Namely, if $H : y^2 = f(x)$ is an hyperelliptic curve, with $\deg f = 6$, we can associate a canonical basis of differentials $(dx/y, xdx/y)$ on $\mathrm{Jac}(H)$, via the canonical isomorphism $\Omega^1(\mathrm{Jac}(C)) \simeq H^1(C)$. Then the curve equation gives a universal vectorial modular $(\mathrm{Jac}(H), dx/y, xdx/y) \mapsto f(x)$ form of weight Sym^6. If (j_1, j_2, j_3) are the Igusa invariants, we can thus express the vectorial modular functions dj_1, dj_2, dj_3 of weight Sym^2 in terms of the coefficients of the curve equation $f(x)$: explicit formulas are in [KPR24].

We can then proceed as in Section 3.3 to reconstruct an isogeny from the modular polynomial: letting $J = (j_1, j_2, j_3)$, and Φ_ℓ the modular polynomial(s) in dimension 2, if $\Phi_\ell(J(A), J(B)) = 0$, differentiatig Φ_ℓ gives the relationship between $dJ(A, \omega_A)$ and $dJ(B, \omega_B)$ where ω_A, ω_B are normalised basis of differentials on A, B, i.e., if $\phi : A \to B$ is the associated ℓ-isogeny, we have $\phi^* \omega_B = \omega_A$. Using the formulas expressing dJ in terms of the curve equation, we can find equations for H_A, H_B such that the isogeny is normalised with respect to the canonical basis $(dx/y, xdx/y)$. We can then solve a differential equation to recover ϕ in Mumford coordinates. See [Rob21, § 5.4.2] for a summary, and [KPR24] for all details.

In theory, the heat equation

$$2\pi i(1 + \delta_{jk}) \frac{\partial \theta_i}{\partial \tau_{jk}} = \frac{\partial^2 \theta_i}{\partial z_j \partial z_k}.$$

gives the Kodaira-Spencer isomorphism for theta functions. So we could extend the approach of [KPR24] to the theta model in higher dimension. The main problem is that the size of the modular polynomial explodes in higher dimension (see [Kie22b] for bounds on the degree and height). For instance, the Siegel modular polynomial in dimension g is of size $\widetilde{O}(\ell^{N(N+2)})$ where $N = g(g+1)/2$ is the dimension of the moduli space $\mathcal{A}_g$; and the evaluated modular polynomial $\Phi_\ell(J(A), Y)$ for $A/\mathbb{F}_q$ is of size $\widetilde{O}(\ell^N \log q)$. When $g = 1$, we recover that ϕ_ℓ is of size $\widetilde{O}(\ell^3)$, and the evaluated modular polynomial of size $O(\ell \log q)$. In dimension 2, the Siegel modular polynomial Φ_ℓ is of size $\widetilde{O}(\ell^{15})$, and the evaluated modular polynomial is of size $O(\ell^3 \log q)$, but the best algorithm we have to evaluate it [Kie20] uses analytic method and cost time $O(\ell^6 \log q)$, if $q = p$ (see also [Rob22b] for CRT and p-adic variants which use the HD representation). In dimension 3, the Siegel modular polynomial Φ_ℓ is of a staggering size $\widetilde{O}(\ell^{48})$, and even the evaluated modular polynomial of size $O(\ell^6 \log q)$.

In higher dimension, using Hilbert modular polynomials to parametrize β-isogenies, with $N(\beta) = \ell$, is more reasonable: they take space $\widetilde{O}(\ell^{g+2})$ and their evaluation take space $\widetilde{O}(\ell \log q)$. We refer to [Kie22a; Kie21] for applications to point counting in dimension 2.

References

[ADMP20] N. Alamati, L. De Feo, H. Montgomery, and S. 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 26*. Springer. 2020, pp. 411–439 (cit. on p. 23).

[AHHR24] J. Alper, J. Hall, D. Halpern-Leistner, and D. Rydh. "Artin algebraization for pairs with applications to the local structure of stacks and Ferrand pushouts". In: *Forum of Mathematics, Sigma*. Vol. 12. Cambridge University Press. 2024, e20 (cit. on p. 57).

[And17] Y. André. "On the Kodaira–Spencer map of abelian schemes". In: *Ann. Sc. Norm. Super. Pisa Cl. Sci. (5)* 17.4 (2017), pp. 1397–1416 (cit. on p. 58).

[Arp22] S. Arpin. "Adding level structure to supersingular elliptic curve isogeny graphs". In: *arXiv preprint arXiv:2203.03531* (2022) (cit. on pp. 50, 58).

[ACL+23] S. Arpin, C. Camacho-Navarro, K. Lauter, J. Lim, K. Nelson, T. Scholl, and J. Sotáková. "Adventures in supersingularland". In: *Experimental Mathematics* 32.2 (2023), pp. 241–268 (cit. on p. 21).

[ACL+22] S. Arpin, M. Chen, K. E. Lauter, R. Scheidler, K. E. Stange, and H. T. Tran. "Orientations and cycles in supersingular isogeny graphs". In: *Proceedings of Women in Number Theory* 5 (2022) (cit. on p. 21).

[ACD+23] S. Arpin, J. Clements, P. Dartois, J. K. Eriksen, P. Kutas, and B. Wesolowski. "Finding orientations of supersingular elliptic curves and quaternion orders". In: *arXiv preprint arXiv:2308.11539* (2023) (cit. on p. 1).

[BGDS23] G. Banegas, V. Gilchrist, A. L. Dévéhat, and B. Smith. "Fast and Frobenius: Rational Isogeny Evaluation over Finite Fields". In: *International Conference on Cryptology and Information Security in Latin America*. Springer. 2023, pp. 129–148 (cit. on pp. 18, 45).

[Bas24] A. Basso. "POKE: A Framework for Efficient PKEs, Split KEMs, and OPRFs from Higher-dimensional Isogenies". In: *Cryptology ePrint Archive* (2024) (cit. on pp. 1, 27, 40).

[BCC+23] A. Basso, G. Codogni, D. Connolly, L. De Feo, T. B. Fouotsa, G. M. Lido, T. Morrison, L. Panny, S. Patranabis, and B. Wesolowski. "Supersingular curves you can trust". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2023, pp. 405–437 (cit. on pp. 27, 40).

[BDD+24] A. Basso, L. De Feo, P. Dartois, A. Leroux, L. Maino, G. Pope, D. Robert, and B. Wesolowski. "SQIsign2D-West: The Fast, the Small, and the Safer". Accepted for publication at Asiacrypt 2024. Aug. 2024 (cit. on pp. 1, 22, 30, 35, 39, 53, 54, 56).

[BMP23] A. Basso, L. Maino, and G. Pope. "FESTA: fast encryption from supersingular torsion attacks". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2023, pp. 98–126 (cit. on pp. 1, 27).

[BDLS20] D. Bernstein, L. De Feo, A. Leroux, and B. Smith. "Faster computation of isogenies of large prime degree". In: *Algorithmic Number Theory Symposium (ANTS XIV)*. Vol. 4. 1. Mathematical Sciences Publishers, 2020, pp. 39–55. arXiv: 2003.10118. URL: https://msp.org/obs/2020/4/p04.xhtml (cit. on pp. 5, 11, 31).

[BDGP23] W. Beullens, L. De Feo, S. D. Galbraith, and C. Petit. "Proving knowledge of isogenies: a survey". In: *Designs, Codes and Cryptography* 91.11 (2023), pp. 3425–3456 (cit. on p. 41).

[BKV19] W. Beullens, T. Kleinjung, and F. Vercauteren. "CSI-FiSh: efficient isogeny based signatures through class group computations". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2019, pp. 227–247 (cit. on pp. 23, 24).

[BM21] B. Bhatt and A. Mathew. "The arc-topology". In: *Duke Mathematical Journal* 170.9 (2021), pp. 1899–1988 (cit. on p. 57).

[BCR10] G. Bisson, R. Cosset, and D. Robert. *AVIsogenies*. Magma package devoted to the computation of isogenies between abelian varieties. 2010. URL: https://www.math.u-bordeaux.fr/~damienrobert/avisogenies/. Free software (LGPLv2+), registered to APP (reference IDDN.FR.001.440011.000.R.P.2010.000.10000). Latest version 0.7, released on 2021-03-13. (Cit. on p. 28).

[BMSS08] A. Bostan, F. Morain, B. Salvy, and E. Schost. "Fast algorithms for computing isogenies between elliptic curves". In: *Mathematics of Computation* 77.263 (2008), pp. 1755–1778 (cit. on pp. 13, 16).

[BCG+17] A. Bostan, F. Chyzak, M. Giusti, R. Lebreton, G. Lecerf, B. Salvy, and É. Schost. *Algorithmes efficaces en calcul formel*. Published by the authors, 2017. URL: https://hal.inria.fr/hal-01431717/document (cit. on pp. 6, 12).

[BDDFS19] L. Brieulle, L. De Feo, J. Doliskani, J.-P. Flori, and É. Schost. "Computing isomorphisms and embeddings of finite fields". In: *Mathematics of Computation* 88.317 (2019), pp. 1391–1426 (cit. on p. 44).

[BLS12] R. Bröker, K. Lauter, and A. Sutherland. "Modular polynomials via isogeny volcanoes". In: *Mathematics of Computation* 81.278 (2012), pp. 1201–1231. arXiv: 1001.0402 (cit. on p. 16).

[BFT14] N. Bruin, E. V. Flynn, and D. Testa. "Descent via (3, 3)-isogeny on Jacobians of genus 2 curves". In: *Acta Arithmetica* 165.3 (2014), pp. 201–223 (cit. on p. 29).

[BSC+23] G. Bruno, M. C.-R. Santos, C. Costello, J. K. Eriksen, M. Meyer, M. Naehrig, and B. Sterner. "Cryptographic smooth neighbors". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2023, pp. 190–221 (cit. on p. 52).

[CF+96] J. W. S. Cassels, E. V. Flynn, et al. *Prolegomena to a middlebrow arithmetic of curves of genus 2*. Vol. 230. Cambridge University Press, 1996 (cit. on pp. 29, 31).

[CD21] W. Castryck and T. Decru. "Multiradical isogenies". In: *Arithmetic, Geometry, Cryptography, and Coding Theory* 779 (2021), pp. 57–89 (cit. on p. 12).

[CD22] W. Castryck and T. Decru. *An efficient key recovery attack on SIDH (preliminary version)*. Cryptology ePrint Archive, Paper 2022/975. 2022. URL: https://eprint.iacr.org/2022/975 (cit. on p. 1).

[CD23] W. Castryck and T. Decru. "An efficient key recovery attack on SIDH". In: Springer-Verlag (Eurocrypt 2023), Apr. 2023, pp. 423–447. DOI: 10.1007/978-3-031-30589-4_15 (cit. on pp. 1, 2, 26).

[CDHV22] W. Castryck, T. Decru, M. Houben, and F. Vercauteren. "Horizontal racewalking using radical isogenies". In: *International Conference on the Theory and Application of Cryptology and Information Security (Asiacrypt)*. Springer. 2022, pp. 67–96 (cit. on p. 12).

[CDM+24] W. Castryck, T. Decru, L. Maino, C. Martindale, L. Panny, G. Pope, D. Robert, and B. Wesolowski. "Interpolating isogenies and breaking the SIDH cryptosystem". June 2024 (cit. on pp. 7, 26).

[CDV20] W. Castryck, T. Decru, and F. Vercauteren. "Radical isogenies". In: *International Conference on the Theory and Application of Cryptology and Information Security (Asiacrypt)*. Lecture Notes in Computer Sciencie 12492. Springer. 2020, pp. 493–519 (cit. on p. 12).

[CLMPR18] W. Castryck, T. Lange, C. Martindale, L. Panny, and J. Renes. "CSIDH: an efficient post-quantum commutative group action". In: *International Conference on the Theory and Application of Cryptology and Information Security (Asiacrypt 2018)*. Springer. 2018, pp. 395–427 (cit. on pp. 19, 23).

[CLG09] D. Charles, K. Lauter, and E. Goren. "Cryptographic hash functions from expander graphs". In: *Journal of Cryptology* 22.1 (2009), pp. 93–113. ISSN: 0933-2790 (cit. on p. 11).

[CII+23] M. Chen, M. Imran, G. Ivanyos, P. Kutas, A. Leroux, and C. Petit. "Hidden stabilizers, the isogeny to endomorphism ring problem and the cryptanalysis of pSIDH". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2023, pp. 99–130 (cit. on p. 43).

[CLP24] M. Chen, A. Leroux, and L. Panny. "SCALLOP-HD: group action from 2-dimensional isogenies". In: *IACR International Conference on Public-Key Cryptography*. Springer. 2024, pp. 190–216 (cit. on pp. 1, 8, 25, 52).

[CS21] M. Chenu and B. Smith. "Higher-degree supersingular group actions". In: *arXiv preprint arXiv:2107.08832* (2021) (cit. on p. 20).

[CC86] D. Chudnovsky and G. Chudnovsky. "Sequences of numbers generated by addition in formal groups and new primality and factorization tests". In: *Advances in Applied Mathematics* 7.4 (1986), pp. 385–434. ISSN: 0196-8858. DOI: https://doi.org/10.1016/0196-8858(86)90023-0 (cit. on p. 30).

[CL23] G. Codogni and G. Lido. "Spectral theory of isogeny graphs". In: *arXiv preprint arXiv:2308.13913* (2023) (cit. on p. 27).

[CK20] L. Colo and D. Kohel. "Orienting supersingular isogeny graphs". In: *Journal of Mathematical Cryptology* 14.1 (2020), pp. 414–437 (cit. on p. 20).

[Cor07] G. Cornacchia. *Su di un metodo per la risoluzione in numeri interi dell'equazione* $\sum_{h=0}^{n} C_h x^{n-h} y^h = P$. Vol. 46. 1907, pp. 33–90 (cit. on p. 34).

[CCS24] M. Corte-Real Santos, C. Costello, and B. Smith. "Efficient $(3,3)$-isogenies on fast Kummer surfaces". In: *arXiv preprint arXiv:2402.01223* (2024) (cit. on pp. 29, 30).

[CEMR24] M. Corte-Real Santos, J. K. Eriksen, M. Meyer, and K. Reijnders. "AprésSQI: extra fast verification for SQIsign using extension-field signing". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2024, pp. 63–93 (cit. on pp. 46, 54).

[CR24] M. Corte-Real Santos and K. Reijnders. *Return of the Kummer: a toolbox for genus 2 cryptography*. Cryptology ePrint Archive, Paper 2024/948. 2024. URL: https://eprint.iacr.org/2024/948 (cit. on p. 31).

[CR15] R. Cosset and D. Robert. "An algorithm for computing (ℓ,ℓ)-isogenies in polynomial time on Jacobians of hyperelliptic curves of genus 2". In: *Mathematics of Computation* 84.294 (Nov. 2015), pp. 1953–1975. DOI: 10.1090/S0025-5718-2014-02899-8 (cit. on pp. 2, 28).

[CMSV19] E. Costa, N. Mascot, J. Sijsling, and J. Voight. "Rigorous computation of the endomorphism ring of a Jacobian". In: *Mathematics of Computation* 88.317 (2019), pp. 1303–1339 (cit. on p. 31).

[Cos18] C. 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 24*. Springer. 2018, pp. 428–456 (cit. on p. 31).

[CJL+17] C. Costello, D. Jao, P. Longa, M. Naehrig, J. Renes, and D. Urbanik. "Efficient compression of SIDH public keys". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2017, pp. 679–706 (cit. on p. 46).

[Cou09] J. Couveignes. "Linearizing torsion classes in the Picard group of algebraic curves over finite fields". In: *Journal of Algebra* 321.8 (2009), pp. 2085–2118. ISSN: 0021-8693 (cit. on p. 45).

[Cou06] J. M. Couveignes. "Hard Homogeneous Spaces." In: *IACR Cryptology ePrint Archive* 2006 (2006), p. 291 (cit. on p. 19).

[CE14] J.-M. Couveignes and T. Ezome. "Computing functions on Jacobians and their quotients". In: *LMS Journal of Computation and Mathematics* 18.1 (2014), pp. 555–577. arXiv: 1409.0481 (cit. on pp. 29, 31).

[Dar24] P. Dartois. "Fast computation of 2-isogenies in dimension 4 with the theta model and cryptographic applications". 2024 (cit. on pp. 30, 48).

[DLRW24] P. Dartois, A. Leroux, D. Robert, and B. Wesolowski. "SQISignHD: New Dimensions in Cryptography". In: Lecture Notes in Computer Science 14651 (May 2024). Ed. by M. Joye and G. Leander, pp. 3–32. DOI: 10.1007/978-3-031-58716-0_1 (cit. on pp. 1, 7, 8, 22, 29, 30, 47, 48, 52, 55).

[DMPR24] P. Dartois, L. Maino, G. Pope, and D. Robert. "An Algorithmic Approach to $(2,2)$-isogenies in the Theta Model and Applications to Isogeny-based Cryptography". Accepted for publication at Asiacrypt 2024. Aug. 2024 (cit. on pp. 2, 29, 30, 48).

[De 17] L. De Feo. *Mathematics of Isogeny Based Cryptography*. 2017. arXiv: 1711.04062 (cit. on p. 20).

[DDGZ22] L. De Feo, S. Dobson, S. D. Galbraith, and L. Zobernig. "SIDH proof of knowledge". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2022, pp. 310–339 (cit. on pp. 40, 41).

[DDS14] L. De Feo, J. Doliskani, and É. Schost. "Fast arithmetic for the algebraic closure of finite fields". In: *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*. 2014, pp. 122–129 (cit. on p. 43).

[DFP24] L. De Feo, T. B. Fouotsa, and L. Panny. "Isogeny problems with level structure". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2024, pp. 181–204 (cit. on pp. 27, 41).

[DG19] L. De Feo and S. D. Galbraith. "SeaSign: compact isogeny signatures from class group actions". 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 III 38*. Springer. 2019, pp. 759–789 (cit. on p. 23).

[DHPS16] L. De Feo, C. Hugounenq, J. Plût, and É. Schost. "Explicit isogenies in quadratic time in any characteristic". In: *LMS Journal of Computation and Mathematics* 19.A (2016), pp. 267–282 (cit. on p. 16).

[DJP14] L. De Feo, D. Jao, and J. Plût. "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies". In: *Journal of Mathematical Cryptology* 8.3 (2014), pp. 209–247 (cit. on pp. 1, 17).

[DKS18] L. De Feo, J. Kieffer, and B. Smith. "Towards practical key exchange from ordinary isogeny graphs". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2018, pp. 365–394. arXiv: 1809.07543 (cit. on p. 19).

[DKLPW20] L. De Feo, D. Kohel, A. Leroux, C. Petit, and B. Wesolowski. "SQISign: compact post-quantum signatures from quaternions and isogenies". In: *International Conference on the Theory and Application of Cryptology and Information Security (Asiacrypt 2020)*. Springer. 2020, pp. 64–93 (cit. on pp. 3, 21, 50, 51, 54).

[DLLW23] L. De Feo, A. Leroux, P. Longa, and B. Wesolowski. "New algorithms for the Deuring correspondence: towards practical and secure SQISign signatures". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2023, pp. 659–690 (cit. on pp. 3, 22, 52, 54).

[Dec24] T. Decru. "Radical Vélu N-Isogeny Formulae". In: *Annual International Cryptology Conference (Eurocrypt)*. Springer. 2024, pp. 107–128 (cit. on p. 12).

[DK23] T. Decru and S. Kunzweiler. "Efficient Computation of (3 n, 3 n)-Isogenies". In: *International Conference on Cryptology in Africa*. Springer. 2023, pp. 53–78 (cit. on p. 30).

[DMS23] T. Decru, L. Maino, and A. Sanso. "Towards a quantum-resistant weak verifiable delay function". In: *International Conference on Cryptology and Information Security in Latin America*. Springer. 2023, pp. 149–168 (cit. on p. 1).

[DHK+23] J. Duman, D. Hartmann, E. Kiltz, S. Kunzweiler, J. Lehmann, and D. Riepel. "Generic models for group actions". In: *IACR International Conference on Public-Key Cryptography*. Springer. 2023, pp. 406–435 (cit. on p. 23).

[DF24] M. Duparc and T. B. Fouotsa. "SQIPrime: A dimension 2 variant of SQISignHD with non-smooth challenge isogenies". In: *Cryptology ePrint Archive* (2024) (cit. on pp. 1, 22, 30, 35, 56).

[DFV24] M. Duparc, T. B. Fouotsa, and S. Vaudenay. "Silbe: an updatable public key encryption scheme from lollipop attacks". In: *Cryptology ePrint Archive* (2024) (cit. on p. 1).

[EGM12] B. Edixhoven, G. van der Geer, and B. Moonen. *Abelian varieties*. Book project, 2012. URL: http://van-der-geer.nl/~gerard/AV.pdf (cit. on p. 58).

[ES24] K. Eisentraeger and G. Scullard. "Connecting Kani's Lemma and path-finding in the Bruhat-Tits tree to compute supersingular endomorphism rings". In: (2024). arXiv: 2402.05059 (cit. on pp. 1, 37).

[EHLMP18] K. Eisenträger, S. Hallgren, K. Lauter, T. Morrison, and C. Petit. "Supersingular isogeny graphs and endomorphism rings: reductions and solutions". In: *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 37. Springer. 2018, pp. 329–368 (cit. on p. 21).

[Elk92] N. Elkies. "Explicit isogenies". In: *manuscript, Boston MA* (1992) (cit. on p. 16).

[Elk97] N. Elkies. "Elliptic and modular curves over finite fields and related computational issues". In: *Computational perspectives on number theory: proceedings of a conference in honor of AOL Atkin, September 1995, University of Illinois at Chicago*. Vol. 7. Amer Mathematical Society. 1997, p. 21 (cit. on p. 16).

[EHIK21] E. Elmanto, M. Hoyois, R. Iwasa, and S. Kelly. "Cdh descent, cdarc descent, and Milnor excision". In: *Mathematische Annalen* 379.3 (2021), pp. 1011–1045 (cit. on p. 57).

[Eng09] A. Enge. "Computing modular polynomials in quasi-linear time". In: *Math. Comp* 78.267 (2009), pp. 1809–1824 (cit. on p. 16).

[EPSV] J. K. Eriksen, L. Panny, J. Sotáková, and M. Veroni. "Deuring for the people: Supersingular elliptic curves with prescribed endomorphism ring in general characteristic". In: *Contempory Mathematics* 796 (). LuCaNT: LMFDB, computation, and number theory (Providence), Proceedings, pp. 339–373 (cit. on pp. 18, 46, 50).

[FC90] G. Faltings and C.-L. Chai. *Degeneration of abelian varieties*. Ergebnisse der Mathematik und ihrer Grenzgebiete (3) 22. Springer-Verlag, Berlin, 1990 (cit. on p. 58).

[Feo10] L. de Feo. "Algorithmes Rapides pour les Tours de Corps Finis et les Isogénies". PhD thesis. Ecole Polytechnique X, Dec. 2010. URL: http://hal.inria.fr/tel-00547034/en (cit. on p. 10).

[FFK+23] L. D. Feo, T. B. Fouotsa, P. Kutas, A. Leroux, S.-P. Merz, L. Panny, and B. Wesolowski. "SCALLOP: scaling the CSI-FiSh". In: *IACR International Conference on Public-Key Cryptography*. Springer. 2023, pp. 345–375 (cit. on pp. 20, 25).

[Fly15] E. V. Flynn. "Descent via (5, 5)-isogeny on Jacobians of genus 2 curves". In: *Journal of Number Theory* 153 (2015), pp. 270–282 (cit. on p. 29).

[Fly93] E. V. Flynn. "The group law on the jacobian of a curve of genus 2." In: *Journal für die reine und angewandte Mathematik* 439 (1993), pp. 45–69 (cit. on p. 29).

[FMP23] T. B. Fouotsa, T. Moriya, and C. Petit. "M-SIDH and MD-SIDH: countering SIDH attacks by masking information". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2023, pp. 282–309 (cit. on p. 27).

[FM02] M. Fouquet and F. Morain. "Isogeny volcanoes and the SEA algorithm". In: *Algorithmic number theory (Sydney, 2002)*. Vol. 2369. Lecture Notes in Comput. Sci. Berlin: Springer, 2002, pp. 276–291. DOI: 10.1007/3-540-45455-1_23 (cit. on p. 18).

[FK11] G. Frey and E. Kani. "Correspondences on hyperelliptic curves and applications to the discrete logarithm". In: *International Joint Conferences on Security and Intelligent Information Systems*. Springer. 2011, pp. 1–19 (cit. on p. 29).

[Gal24] S. Galbraith. *Climbing and descending tall volcanos*. Cryptology ePrint Archive, Paper 2024/924. 2024. URL: https://eprint.iacr.org/2024/924 (cit. on pp. 1, 43).

[Gau07] P. Gaudry. "Fast genus 2 arithmetic based on Theta functions". In: *Journal of Mathematical Cryptology* 1.3 (2007), pp. 243–265 (cit. on p. 30).

[GL09] P. Gaudry and D. Lubicz. "The arithmetic of characteristic 2 Kummer surfaces and of elliptic Kummer lines". In: *Finite Fields and Their Applications* 15.2 (2009), pp. 246–260 (cit. on p. 28).

[GPV24] W. Ghantous, F. Pintore, and M. Veroni. "Efficiency of SIDH-based signatures (yes, SIDH)". In: *Journal of Mathematical Cryptology* 18.1 (2024), p. 20230023 (cit. on p. 40).

[HR19] H. Hisil and J. Renes. "On kummer lines with full rational 2-torsion and their usage in cryptography". In: *ACM Transactions on Mathematical Software (TOMS)* 45.4 (2019), pp. 1–17 (cit. on p. 30).

[Ill79] L. Illusie. "Complexe de de Rham-Witt et cohomologie cristalline". In: *Annales scientifiques de l'École Normale Supérieure*. Vol. 12. 4. 1979, pp. 501–661 (cit. on p. 15).

[JD11] D. Jao and L. De Feo. "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies". In: *International Workshop on Post-Quantum Cryptography (PQCrypto 2011)*. Springer. 2011, pp. 19–34 (cit. on p. 1).

[JKP+18] B. W. Jordan, A. G. Keeton, B. Poonen, E. M. Rains, N. Shepherd-Barron, and J. T. Tate. "Abelian varieties isogenous to a power of an elliptic curve". In: *Compositio Mathematica* 154.5 (2018), pp. 934–959 (cit. on p. 20).

[Kan97] E. Kani. "The number of curves of genus two with elliptic differentials." In: *Journal für die reine und angewandte Mathematik* 485 (1997), pp. 93–122 (cit. on pp. 1, 7, 32).

[Kan11] E. Kani. "Products of CM elliptic curves". In: *Collectanea mathematica* 62.3 (2011), pp. 297–339 (cit. on p. 19).

[KU11] K. S. Kedlaya and C. Umans. "Fast polynomial factorization and modular composition". In: *SIAM Journal on Computing* 40.6 (2011), pp. 1767–1802 (cit. on pp. 18, 31, 44, 45).

[Kem88] G. Kempf. "Multiplication over abelian varieties". In: *American Journal of Mathematics* 110.4 (1988), pp. 765–773 (cit. on p. 28).

[Kem89a] G. Kempf. "Linear systems on abelian varieties". In: *American Journal of Mathematics* 111.1 (1989), pp. 65–94 (cit. on p. 28).

[Kem92] G. Kempf. "Equations of Kümmer Varieties". In: *American Journal of Mathematics* 114.1 (1992), pp. 229–232 (cit. on p. 28).

[Kem89b] G. Kempf. "Projective coordinate rings of abelian varieties". In: *Algebraic analysis, geometry and number theory* (1989), pp. 225–236 (cit. on p. 28).

[Kem90] G. R. Kempf. "Some wonderful rings in algebraic geometry". In: *Journal of Algebra* 134.1 (1990), pp. 222–224 (cit. on p. 28).

[Kie20] J. Kieffer. "Evaluating modular polynomials in genus 2". 2020. arXiv: 2010.10094 [math.NT]. HAL: hal-02971326. (Cit. on pp. 16, 59).

[Kie21] J. Kieffer. "Higher-dimensional modular equations, applications to isogeny computations and point counting". Thèse de doctorat dirigée par Damien Robert, Mathématiques Pures, Université de Bordeaux. PhD thesis. 2021. URL: http://www.theses.fr/2021BORD0188 (cit. on p. 59).

[Kie22a] J. Kieffer. "Counting points on abelian surfaces over finite fields with Elkies's method". In: (2022). arXiv: 2203.02009 [math.NT] (cit. on p. 59).

[Kie22b] J. Kieffer. "Degree and height estimates for modular equations on PEL Shimura varieties". 2022 (cit. on p. 59).

[KPR24] J. Kieffer, A. Page, and D. Robert. "Computing isogenies from modular equations between Jacobians of genus 2 curves". Accepted for publication at Journal of Algebra. June 2024. arXiv: 2001.04137 [math.AG] (cit. on pp. 14, 16, 31, 59).

[KR22] J. Kieffer and D. Robert. "Fast evaluation of modular polynomials and compact representation of isogenies between elliptic curves". Aug. 2022 (cit. on p. 13).

[KNRR21] M. Kirschmer, F. Narbonne, C. Ritzenthaler, and D. Robert. "Spanning the isogeny class of a power of an elliptic curve". In: *Mathematics of Computation* 91.333 (Sept. 2021), pp. 401–449. DOI: 10.1090/mcom/3672. arXiv: 2004.08315 (cit. on p. 16).

[Koh96] D. Kohel. "Endomorphism rings of elliptic curves over finite fields". PhD thesis. University of California, 1996 (cit. on pp. 10, 18, 19).

[KLPT14] D. Kohel, K. Lauter, C. Petit, and J.-P. Tignol. "On the quaternion-isogeny path problem". In: *LMS Journal of Computation and Mathematics* 17.A (2014), pp. 418–432 (cit. on pp. 3, 7, 21, 50).

[Kun22] S. Kunzweiler. "Efficient Computation of $(2^n, 2^n)$-Isogenies". In: *Cryptology ePrint Archive* (2022) (cit. on p. 29).

[KR24] S. Kunzweiler and D. Robert. "Computing modular polynomials by deformation". Accepted for publication at ANTS XVI Conference. June 2024 (cit. on pp. 1, 16, 38).

[Kup05] G. Kuperberg. "A subexponential-time quantum algorithm for the dihedral hidden subgroup problem". In: *SIAM Journal on Computing* 35.1 (2005), pp. 170–188 (cit. on p. 19).

[LS08] R. Lercier and T. Sirvent. "On Elkies subgroups of ℓ-torsion points in elliptic curves defined over a finite field." In: *Journal de théorie des nombres de Bordeaux* 20.3 (2008), pp. 783–797 (cit. on p. 13).

[Ler22a] A. Leroux. "A new isogeny representation and applications to cryptography". In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2022, pp. 3–35 (cit. on pp. 4, 7, 22).

[Ler22b] A. Leroux. "Quaternion algebras and isogeny-based cryptography ". PhD thesis. LIX, 2022 (cit. on pp. 19, 22, 39, 50, 51, 58).

[Ler23a] A. Leroux. "Computation of Hilbert class polynomials and modular polynomials from supersingular elliptic curves". In: *Cryptology ePrint Archive* (2023) (cit. on p. 16).

[Ler23b] A. Leroux. "Verifiable random function from the Deuring correspondence and higher dimensional isogenies". In: (2023) (cit. on pp. 1, 8, 22, 52).

[LR10] D. Lubicz and D. Robert. "Efficient pairing computation with theta functions". In: ed. by G. Hanrot, F. Morain, and E. Thomé. Vol. 6197. Lecture Notes in Computer Science. 9th International Symposium, Nancy, France, ANTS-IX, July 19-23, 2010, Proceedings. Springer–Verlag, July 2010. DOI: 10.1007/978-3-642-14518-6_21 (cit. on p. 28).

[LR12] D. Lubicz and D. Robert. "Computing isogenies between abelian varieties". In: *Compositio Mathematica* 148.5 (Sept. 2012), pp. 1483–1515. DOI: 10.1112/S0010437X12000243. arXiv: 1001.2016 [math.AG] (cit. on p. 28).

[LR15a] D. Lubicz and D. Robert. "A generalisation of Miller's algorithm and applications to pairing computations on abelian varieties". In: *Journal of Symbolic Computation* 67 (Mar. 2015), pp. 68–92. DOI: 10.1016/j.jsc.2014.08.001 (cit. on p. 28).

[LR15b] D. Lubicz and D. Robert. "Computing separable isogenies in quasi-optimal time". In: *LMS Journal of Computation and Mathematics* 18 (1 Feb. 2015), pp. 198–216. DOI: 10.1112/S146115701400045X. arXiv: 1402.3628 (cit. on pp. 28, 31).

[LR16] D. Lubicz and D. Robert. "Arithmetic on Abelian and Kummer Varieties". In: *Finite Fields and Their Applications* 39 (May 2016), pp. 130–158. DOI: 10.1016/j.ffa.2016.01.009 (cit. on p. 28).

[LR22] D. Lubicz and D. Robert. "Fast change of level and applications to isogenies". In: *Research in Number Theory (ANTS XV Conference)* 9.1 (Dec. 2022). DOI: 10.1007/s40993-022-00407-9 (cit. on pp. 7, 28).

[MM22] L. Maino and C. Martindale. *An attack on SIDH with arbitrary starting curve*. Cryptology ePrint Archive, Paper 2022/1026. 2022. URL: https://eprint.iacr.org/2022/1026 (cit. on p. 2).

[MMPPW23] L. Maino, C. Martindale, L. Panny, G. Pope, and B. Wesolowski. "A direct key recovery attack on SIDH". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2023, pp. 448–471 (cit. on pp. 1, 2, 26, 32).

[MW23] A. H. L. Merdy and B. Wesolowski. "The supersingular endomorphism ring problem given one endomorphism". In: *arXiv preprint arXiv:2309.11912* (2023) (cit. on pp. 1, 37).

[Mil20] E. Milio. "Computing isogenies between Jacobians of curves of genus 2 and 3". In: *Mathematics of Computation* 89.323 (2020), pp. 1331–1364. arXiv: 1709.06063 (cit. on p. 29).

[MR19] E. Milio and D. Robert. "Denominators of modular polynomials on Hilbert surfaces". June 2019 (cit. on p. 2).

[Mor23] T. Moriya. "IS-CUBE: An isogeny-based compact KEM using a boxed SIDH diagram". In: *Cryptology ePrint Archive* (2023) (cit. on pp. 1, 27).

[Mor24] T. Moriya. "LIT-SiGamal: An efficient isogeny-based PKE based on a LIT diagram". In: *Cryptology ePrint Archive* (2024) (cit. on pp. 1, 27).

[Mum66] D. Mumford. "On the equations defining abelian varieties. I". In: *Invent. Math.* 1 (1966), pp. 287–354 (cit. on p. 28).

[Mum67a] D. Mumford. "On the equations defining abelian varieties. II". In: *Invent. Math.* 3 (1967), pp. 75–135 (cit. on p. 28).

[Mum67b] D. Mumford. "On the equations defining abelian varieties. III". In: *Invent. Math.* 3 (1967), pp. 215–244 (cit. on p. 28).

[Mum83] D. Mumford. *Tata lectures on theta I*. Vol. 28. Progress in Mathematics. With the assistance of C. Musili, M. Nori, E. Previato and M. Stillman. Boston, MA: Birkhäuser Boston Inc., 1983, pp. xiii+235. ISBN: 3-7643-3109-7 (cit. on p. 28).

[Mum84] D. Mumford. *Tata lectures on theta II*. Vol. 43. Progress in Mathematics. Jacobian theta functions and differential equations, With the collaboration

of C. Musili, M. Nori, E. Previato, M. Stillman and H. Umemura. Boston, MA: Birkhäuser Boston Inc., 1984, pp. xiv+272. ISBN: 0-8176-3110-0 (cit. on p. 28).

[Mum91] D. Mumford. *Tata lectures on theta III*. Vol. 97. Progress in Mathematics. With the collaboration of Madhav Nori and Peter Norman. Boston, MA: Birkhäuser Boston Inc., 1991, pp. viii+202. ISBN: 0-8176-3440-1 (cit. on p. 28).

[NO23] K. Nakagawa and H. Onuki. "QFESTA: Efficient Algorithms and Parameters for FESTA using Quaternion Algebras". In: *Cryptology ePrint Archive* (2023) (cit. on pp. 1, 8, 39).

[NO24] K. Nakagawa and H. Onuki. "SQIsign2D-East: A New Signature Scheme Using 2-dimensional Isogenies". In: *Cryptology ePrint Archive* (2024) (cit. on pp. 1, 22, 30, 35, 39, 56).

[Nic18] C. Nicholls. "Descent methods and torsion on Jacobians of higher genus curves". PhD thesis. University of Oxford, 2018 (cit. on p. 29).

[Oda69] T. Oda. "The first de Rham cohomology group and Dieudonné modules". In: *Annales scientifiques de l'École Normale Supérieure*. Vol. 2. 1. 1969, pp. 63–135 (cit. on pp. 13, 15).

[Onu21] H. Onuki. "On oriented supersingular elliptic curves". In: *Finite Fields and Their Applications* 69 (2021), p. 101777 (cit. on p. 20).

[OM22] H. Onuki and T. Moriya. "Radical isogenies on Montgomery curves". In: *IACR International Conference on Public-Key Cryptography*. Lecture Notes in Computer Science 13177. Springer. 2022, pp. 473–497 (cit. on p. 12).

[ON24] H. Onuki and K. Nakagawa. "Ideal-to-isogeny algorithm using 2-dimensional isogenies and its application to SQIsign". In: *Cryptology ePrint Archive* (2024) (cit. on pp. 1, 22, 52).

[PR23a] A. Page and D. Robert. "Clapotis: Evaluating the isogeny class group action in polynomial time". Nov. 2023 (cit. on pp. 20, 24, 58).

[PR23b] A. Page and D. Robert. "Introducing Clapoti(s): Evaluating the isogeny class group action in polynomial time". Nov. 2023 (cit. on pp. 1, 7, 8, 21–23, 39, 53).

[PW24] A. Page and B. Wesolowski. "The supersingular endomorphism ring and one endomorphism problems are equivalent". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2024, pp. 388–417 (cit. on pp. 1, 27, 37).

[Pei20] C. Peikert. "He gives C-sieves on the CSIDH". In: *Annual international conference on the theory and applications of cryptographic techniques*. Springer. 2020, pp. 463–492 (cit. on p. 19).

[PT18] P. Pollack and E. Treviño. "Finding the Four Squares in Lagrange's Theorem." In: *Integers* 18 (2018), A15 (cit. on p. 34).

[Pri24] V. Pribanic. "Radical isogenies and modular curves." In: *Advances in Mathematics of Communications* 18 (2024), pp. 1748–1767 (cit. on p. 12).

[RS86] M. O. Rabin and J. O. Shallit. "Randomized algorithms in number theory". In: *Communications on Pure and Applied Mathematics* 39.S1 (1986), S239–S256 (cit. on p. 34).

[RSSB16] J. Renes, P. Schwabe, B. Smith, and L. Batina. "μKummer: Efficient Hyperelliptic Signatures and Key Exchange on Microcontrollers". In: *International Conference on Cryptographic Hardware and Embedded Systems*. Lecture

Notes in Computer Science 9813. Springer. 2016, pp. 301–320. DOI: 10.1007/978-3-662-53140-2_15 (cit. on p. 30).

[Ric36] F. Richelot. "Essai sur une méthode générale pour déterminer la valeur des intégrales ultra-elliptiques, fondée sur des transformations remarquables de ces transcendantes". In: *C. R. Acad. Sci. Paris* 2 (1836), pp. 622–627 (cit. on p. 29).

[Ric37] F. Richelot. "De transformatione Integralium Abelianorum primiordinis commentation". In: *J. reine angew. Math.* 16 (1837), pp. 221–341 (cit. on p. 29).

[Rob21] D. Robert. "Efficient algorithms for abelian varieties and their moduli spaces". HDR thesis. Université Bordeaux, June 2021. URL: http://www.normalesup.org/~robert/pro/publications/academic/hdr.pdf. Slides: 2021-06-HDR-Bordeaux.pdf (1h, Bordeaux). (Cit. on pp. 2, 7, 13, 16, 28, 29, 31, 45, 59).

[Rob22a] D. Robert. "Evaluating isogenies in polylogarithmic time". Aug. 2022 (cit. on pp. 3, 8, 26, 34).

[Rob22b] D. Robert. "Some applications of higher dimensional isogenies to elliptic curves (overview of results)". Dec. 2022 (cit. on pp. 1, 8, 16, 34, 37, 38, 59).

[Rob23a] D. Robert. "A note on optimising 2^n-isogenies in higher dimension". June 2023. URL: http://www.normalesup.org/~robert/pro/publications/notes/2023-06-optimising_isogenies.pdf (cit. on pp. 29, 30).

[Rob23b] D. Robert. "Breaking SIDH in polynomial time". In: *Eurocrypt 2023* (Apr. 2023). Ed. by C. Hazay and M. Stam, pp. 472–503. DOI: 10.1007/978-3-031-30589-4_17 (cit. on pp. 1–3, 7, 26, 27, 32, 47).

[Rob23c] D. Robert. "The geometric interpretation of the Tate pairing and its applications". Feb. 2023 (cit. on p. 12).

[RS06] A. Rostovtsev and A. Stolbunov. "Public-key cryptosystem based on isogenies". In: *International Association for Cryptologic Research. Cryptology ePrint Archive* (2006). eprint: http://eprint.iacr.org/2006/145 (cit. on p. 19).

[Sch95] R. Schoof. "Counting points on elliptic curves over finite fields". In: *J. Théor. Nombres Bordeaux* 7.1 (1995), pp. 219–254 (cit. on p. 14).

[Sho99] V. 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 (cit. on pp. 18, 44).

[Smi08] B. Smith. "Isogenies and the discrete logarithm problem in Jacobians of genus 3 hyperelliptic curves". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2008, pp. 163–180. arXiv: 0806.2995 [math.NT] (cit. on p. 29).

[Stacks] T. Stacks Project Authors. *Stacks Project*. https://stacks.math.columbia.edu. 2018 (cit. on p. 58).

[Sut11] A. Sutherland. "Structure computation and discrete logarithms in finite abelian p-groups". In: *Mathematics of Computation* 80.273 (2011), pp. 477–500 (cit. on p. 45).

[Sut13a] A. Sutherland. "Isogeny volcanoes". In: *The Open Book Series* 1.1 (2013), pp. 507–530 (cit. on p. 18).

[Sut13b] A. Sutherland. "On the evaluation of modular polynomials". In: *The Open Book Series* 1.1 (2013), pp. 531–555 (cit. on p. 16).

[Tat67] J. T. Tate. "p-Divisible groups". In: *Proceedings of a conference on Local Fields*. Springer. 1967, pp. 158–183 (cit. on p. 15).

[Tia20] S. Tian. "Translating the discrete logarithm problem on Jacobians of genus 3 hyperelliptic curves with (ℓ, ℓ, ℓ)-isogenies". 2020. arXiv: 2007.03172 [math.AG] (cit. on p. 29).

[Tia24] S. Tian. "Computing gluing and splitting (ℓ, ℓ)-isogenies". In: *Designs, Codes and Cryptography* (2024), pp. 1–21 (cit. on p. 29).

[UJ18] D. Urbanik and D. Jao. "SoK: The problem landscape of SIDH". In: *Proceedings of the 5th ACM on ASIA Public-Key Cryptography Workshop*. 2018, pp. 53–60 (cit. on p. 5).

[Vél71] J. Vélu. "Isogénies entre courbes elliptiques". In: *Compte Rendu Académie Sciences Paris Série A-B* 273 (1971), A238–A241 (cit. on pp. 10, 11).

[Voi21] J. Voight. *Quaternion algebras*. Springer Nature, 2021 (cit. on p. 19).

[Wat69] W. Waterhouse. "Abelian varieties over finite fields". In: *Ann. Sci. Ecole Norm. Sup* 2.4 (1969), pp. 521–560 (cit. on p. 19).

[Wes22] B. 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 (cit. on pp. 7, 21, 50).

[ZSPDB18] G. H. Zanon, M. A. Simplicio, G. C. Pereira, J. Doliskani, and P. S. Barreto. "Faster isogeny-based compressed key agreement". In: *International Conference on Post-Quantum Cryptography*. Vol. 68. 5. Springer. IEEE, 2018, pp. 688–701 (cit. on p. 46).

[Zar74] J. G. Zarhin. "A remark on endomorphisms of abelian varieties over function fields of finite characteristic". In: *Mathematics of the USSR-Izvestiya* 8.3 (1974), p. 477 (cit. on p. 2).

INRIA Bordeaux–Sud-Ouest, 200 avenue de la Vieille Tour, 33405 Talence Cedex FRANCE
Email address: damien.robert@inria.fr
URL: http://www.normalesup.org/~robert/

Institut de Mathématiques de Bordeaux, 351 cours de la liberation, 33405 Talence cedex FRANCE