KLaPoTi: An asymptotically efficient isogeny group action from 2-dimensional isogenies

Lorenz Panny[1], Christophe Petit[2,3], and Miha Stopar[2,4]

[1] Technische Universität München, Germany
[2] Université libre de Bruxelles, Belgium
[3] University of Birmingham, UK
[4] Ethereum Foundation

lorenz@yx7.cc christophe.f.petit@gmail.com stopar.miha@gmail.com

Abstract. We construct and implement an efficient post-quantum commutative cryptographic group action based on combining the SCALLOP framework for group actions from isogenies of oriented elliptic curves on one hand with the recent Clapoti method for polynomial-time evaluation of the CM group action on elliptic curves on the other. We take advantage of the very attractive performance of $(2^e, 2^e)$-isogenies between products of elliptic curves in the theta coordinate system. To successfully apply Clapoti in dimension 2, it is required to resolve a particular quadratic diophantine norm equation, for which we employ a slight variant of the KLPT algorithm. Our work marks the first practical instantiation of the CM group action for which both the setup as well as the online phase can be computed in (heuristic) polynomial time.

1 Introduction

Isogenies of abelian varieties, in particular elliptic curves, are viewed as one of the main contenders for post-quantum cryptography. In the context of the NIST post-quantum standardization project, two schemes *SIKE* [20] and *SQIsign* [17] have played prominent roles.

The catastrophic break of SIKE in 2022 was far from the end of the history of isogeny-based cryptography; in fact, quite the opposite: Tools arose from the attack which have since revolutionized algorithms used in the field. The performance issues which are often attributed to isogenies when compared to other post-quantum cryptography paradigms are partially alleviated by new algorithmic tools from and with higher-dimensional isogenies.

Generally speaking, isogeny-based cryptography falls into two large conceptual families, depending on whether they make use of the "full" supersingular isogeny graph or of an isogeny graph of *oriented* curves. The former can be seen as being more rigid as a cryptographic design tool, but more conservative

* Author list in alphabetical order; see https://ams.org/profession/leaders/CultureStatement04.pdf. Christophe Petit is partly supported by EPSRC through grant number EP/V011324/1. Date of this document: 2024-11-10.

in terms of security, while the latter offers some more convenient mathematical structure which is useful to protocol designers; however, it also gives rise to a subexponential (but superpolynomial) quantum attack.

In the first family, the structure of the full supersingular isogeny graph is dictated by lattices embedded in certain quaternion algebras. A key algorithmic tool in this context is the Kohel–Lauter–Petit–Tignol algorithm (KLPT) [24], which is capable of solving diophantine norm equations in some of these lattices efficiently, an operation that is required as a subroutine in many cryptographic constructions. Typically, KLPT has been understood to apply only in the "full supersingular graph" setting. In this paper, we will give an application of this algorithm to the oriented setting.

The "Clapoti" algorithm for group-action evaluation [29] solves the scaling issues which were previously inherent in all approaches to construct a true effective group action from "restricted" effective group actions such as CSIDH [8]. It does so by embedding the oriented isogenies to be constructed inside of a higher-dimensional isogeny, similar to the provable polynomial-time version of the SIDH attack [35]. While this solves the problem in theory, running the full version of the algorithm is currently impractical since it requires isogenies of abelian varieties in dimension 4 or even 8. In this work, we show how to make the Clapoti approach work in dimension 2 in an important special case.

The key issue lies in finding two representatives of an imaginary-quadratic ideal class whose norms sum to a power of two. This type of norm equation looks superficially similar to quaternionic norm equations as encountered by the KLPT algorithm, and indeed, we will show that the KLPT algorithm does extend to this setting. We also note that representing integers by this type of norm form can be seen as a generalization of the four-squares theorem.

Contributions. We revisit the Clapoti method to compute the CM group action on oriented elliptic curves, in particular working only with two-power isogenies of 2-dimensional abelian varieties for efficiency. The result is a novel variant of the CSI-FiSh/SCALLOP/SCALLOP-HD family of group actions which fully solves (in the sense of heuristic polynomial time) the scaling issues inherent in those existing constructions, while offering decent practical performance, as demonstrated by our implementation in Rust. The resulting group action, which we call **SCALLOP2D**, can be used to directly instantiate various group-action-based protocols.

We further discuss two optimizations applicable to non interactive key exchange and signatures respectively. More precisely, we show that a pairing computation normally involved in the group action computation can be altogether avoided by choosing a specific initial curve and slightly tweaking the key exchange protocol. Similarly, we show that the canonical representations normally computed in the group action can be replaced by more efficient random representations in a CSI-FiSh-style signature. We expect these optimizations to be more generally useful in other group-action-based protocols.

Concurrent work. After we first presented an overview of these results at the Quantum Safe Workshop held at IBM Research Zürich on May 24, 2024, we were informed that the authors of [29] had previously and independently discovered the same fundamental idea (using KLPT for Clapoti) while working on [29], but had not announced it publicly. Our work indeed starts from this combination (see Section 5) but reaches far beyond: It also includes the design and implementation of a post-quantum cryptographic commutative group action (see Section 6) resulting from the approach, as well as some techniques and optimizations of independent interest.

2 Preliminaries

For an element α of a quadratic field or quaternion algebra, we write $\mathrm{n}(\cdot)$ for its norm, which is defined as $\alpha\overline{\alpha} \in \mathbb{Q}$. Similarly for ideals $\mathcal{I}$, where $\mathrm{n}(\mathcal{I})$ is the non-negative generator in $\mathbb{Z}$ of the principal ideal $\mathcal{I}\overline{\mathcal{I}}$. In addition, for a quadratic or quaternion ideal $\mathcal{I}$ and an element $\alpha \in \mathcal{I}$, we write $\mathrm{n}_I(\alpha) = \mathrm{n}(\alpha)/\mathrm{n}(\mathcal{I}) \in \mathbb{Z}$ for the "reduced" norm of α relative to $\mathcal{I}$.

Throughout, we consider supersingular elliptic curves E defined over $\mathbb{F}_{p^2}$, where $p \geq 5$, such that $E(\mathbb{F}_{p^2}) = E[p+1]$. (This component of the isogeny graph includes the base-changes of supersingular elliptic curves defined over $\mathbb{F}_p$.) We refer to the endomorphism ring of E by $\mathrm{End}(E)$; it is isomorphic to a maximal order in the quaternion algebra $B_{p,\infty}$ over $\mathbb{Q}$ ramified at p at infinity.

The dual of an isogeny $\varphi\colon E \to E'$ is denoted by $\widehat{\varphi}\colon E' \to E$. On the abstract side (imaginary-quadratic or quaternion orders), taking the dual corresponds to conjugation $\alpha \mapsto \overline{\alpha}$.

2.1 The KLPT algorithm

The quaternion algebra $B_{p,\infty}$ can concretely be represented by a 4-dimensional $\mathbb{Q}$-basis $1, \mathbf{i}, \mathbf{j}, \mathbf{ij}$ satisfying the relations $\mathbf{i}^2 = -q$, $\mathbf{j}^2 = -p$, $\mathbf{ji} = \mathbf{ij}$ for some positive integer q. We consider quaternion orders $\mathcal{Q}$ inside $B_{p,\infty}$: They are full-rank subrings. An order $\mathcal{Q}$ is called *special* if it contains a non-scalar element of very small norm.

Given a (one-sided) ideal $\mathcal{I}$ of a maximal order $\mathcal{Q}$, the KLPT algorithm [24] can be used to find an element $\alpha \in \mathcal{I}$ of given (large enough) norm in polynomial time under various heuristic assumptions. This yields another representative $\mathcal{J} = \mathcal{I}\overline{\alpha}/\mathrm{n}(\mathcal{I})$ of norm $\mathrm{n}_{\mathcal{I}}(\alpha)$ in the (left) ideal class of $\mathcal{I}$. When $\mathcal{Q}$ is special, the KLPT algorithm can produce elements of norm larger than $p^{7/2}$, and a later improvement by Petit–Smith [33] reduces this to p^3. There is also a version of KLPT that only relies on GRH, but outputs elements with larger norms [38]. We postpone a description of the KLPT algorithm to Section 5.

2.2 Oriented curves

Let $\mathcal{O}$ be an imaginary-quadratic order. An *$\mathcal{O}$-oriented elliptic curve* is a pair (E, ι) where $\iota\colon \mathcal{O} \hookrightarrow \mathrm{End}(E)$ is an embedding of rings. For ease of notation, we

will in the following assume that $\mathcal{O}$ is given by a fixed generator $a \in \mathcal{O}$, thus $\mathcal{O} = \mathbb{Z}[a]$, and specify the embedding ι by an explicit endomorphism $\tau := \iota(a)$ instead — this is equivalent and reflects more closely how such an orientation is represented in computational practice. The orientation of E by τ is *primitive* if $\mathbb{Q}(\tau) \cap \mathrm{End}(E) = \mathbb{Z}[\tau]$.

It is well-known that the ideal class group of $\mathcal{O}$ acts on the set of primitively $\mathcal{O}$-oriented elliptic curves; see [11, 28]. In all cases, the action of an (invertible, integral) ideal $\mathfrak{a}$ of $\mathcal{O}$ on a curve (E, τ) is given by evaluating its generators as endomorphisms on E, then returning the (oriented) codomain of the isogeny whose kernel is the intersection of the kernels of those endomorphisms: Set $E[\mathfrak{a}] = \bigcap_{\alpha \in \mathfrak{a}} \ker(\alpha) \leq E$, compute $\phi\colon E \to E'$ with kernel $E[\mathfrak{a}]$, and return $\big(E', (\phi\tau\widehat{\phi})/\deg\phi\big)$.[1] For any ideal $\mathfrak{a}$ of an imaginary-quadratic order $\mathcal{O}$ and $\mathcal{O}$-oriented curve (E, τ), we write $\phi_{\mathfrak{a}} : E \to E_{\mathfrak{a}}$ for the corresponding isogeny.

CSIDH [8] is the CM group action on supersingular elliptic curves over $\mathbb{F}_p$ oriented by the Frobenius order $\mathbb{Z}[\sqrt{-p}]$, for well-chosen primes p. In this case it is not necessary to explicitly encode the endomorphism τ as part of the oriented-curve data; it can be taken as the Frobenius $\pi\colon (x, y) \mapsto (x^p, y^p)$.

For a commutative group action $*\colon G \times X \to X$, one considers the following analogues of the classical discrete-logarithm and Diffie–Hellman problems: The *vectorization* problem is to recover $g \in G$ from a pair $(x, g{*}x)$. The *parallelization* problem is to recover $gh * x$ from a triple $(x, g * x, h * x)$.

2.3 Kani's lemma

Kani's lemma [22] as used in contemporary isogeny-based cryptography permits embedding isogenies of elliptic curves into an isogeny of abelian surfaces (or even higher-dimensional abelian varieties). Concretely, a commutative diagram of elliptic-curve isogenies

$$\begin{array}{ccc} E & \xrightarrow{\varphi} & F \\ {\scriptstyle\psi}\downarrow & & \downarrow{\scriptstyle\psi'} \\ E' & \xrightarrow[\varphi']{} & F' \end{array}$$

with $a := \deg(\varphi) = \deg(\varphi')$, $b := \deg(\psi) = \deg(\psi')$, and $\gcd(a, b) = 1$ induces the $(a{+}b)$-isogeny of (principally polarized) abelian surfaces

$$\Phi = \begin{pmatrix} \varphi & \widehat{\psi'} \\ -\psi & \widehat{\varphi'} \end{pmatrix} : \quad E \times F' \longrightarrow F \times E'$$

which can be computed from its kernel

$$\ker\Phi = \left\{ \big(\widehat{\varphi}(R), \psi'(R)\big) \;:\; R \in F[a+b] \right\}.$$

[1] The divisibility of $\phi\tau\widehat{\phi}$ by $\deg\phi$ follows from the fact that $\ker\phi$ is an eigenspace of $\mathcal{O}$ on $E[\deg\phi]$, hence $\phi\tau\widehat{\phi}(E[\deg\phi]) = \phi\tau(\ker\phi) \subseteq \phi(\ker\phi) = \{\infty\}$.

3 Cryptographic instantiations of the CM action

The CM action has been used in several different ways to construct *restricted* effective group actions (CRS [13, 36], CSIDH [8]), where only some sequences of operations can be applied efficiently, as well as (unrestricted) effective group actions (CSI-FiSh [4], SCALLOP [15], SCALLOP-HD [10]), where arbitrary combinations of operations in the group followed by applying the actions remain efficient. In this section we survey these existing instantiations.

3.1 CSI-FiSh

CSI-FiSh [4] is a signature algorithm based on CSIDH group action, i.e., supersingular elliptic curves oriented by $\mathcal{O} = \mathbb{Z}[\sqrt{-p}]$. The signature is based on the Fiat–Shamir paradigm and an underlying identification protocol with soundness error $1/2$, which is repeated λ times to achieve soundness error $2^{-\lambda}$.

In the identification protocol, the prover chooses an $\mathcal{O}$-ideal $\mathfrak{a}$ as secret key, and they deduce their public key $E = \mathfrak{a} * E_0$. Upon starting an interactive proof of knowledge of $\mathfrak{a}$, they choose a random ideal $\mathfrak{b}$ and send a commitment $E_c = \mathfrak{b} * E$. The challenge is made of a single bit, and depending on its value the prover answers with either $\mathfrak{b}$ or an ideal $\mathfrak{c}$ in the equivalence class of $\mathfrak{ba}$. The verifier then verifies whether E_c is $\mathfrak{b} * E$ or $\mathfrak{c} * E_0$ depending on the challenge bit. Note that returning $\mathfrak{ba}$ directly would leak the secret as this random variable has mean $\mathfrak{b}_0\mathfrak{a}$ where $\mathfrak{b}_0$ is the expected value of the random variable $\mathfrak{b}$, but returning a canonical representative in the ideal class of $\mathfrak{ba}$ solves the problem if the class of $\mathfrak{b}$ is (sufficiently close to) uniform. Moreover, this representative must be chosen such that the group action $\mathfrak{c} * E_0$ can be efficiently computed.

In the CSI-FiSh protocol, the security level and characteristic are fixed. A short basis for the *lattice of relations in the class group of* $\mathcal{O}$ is precomputed. Fix some set of small primes ℓ_i that split in $\mathcal{O}$ as $\ell_i\mathcal{O} = \mathfrak{l}_i\overline{\mathfrak{l}_i}$ where $\mathfrak{l}_i$ is some ideal of norm ℓ_i. Then the relation lattice is

$$\mathcal{L}_B = \left\langle (e_2, e_3, e_5, \ldots) \;\middle|\; \prod_{\ell_i < B} \mathfrak{l}_i^{e_i} = 1 \right\rangle.$$

Computing a suitable representative $\mathfrak{c}$ then amounts to computing a reduced representative in the ideal class of $\mathfrak{ba}$, then a B-smooth representative in this class, and finally reducing it by the relation lattice to obtain a canonical powersmooth representative.

The CSI-FiSh paper provides concrete timings for CSIDH-512 parameters based on a record class group (pre-)computation. However, larger parameter sets appear out of practical range, with the best class group computation algorithms requiring $L_p[1/2]$ subexponential time [21, 5, 23, 4]. The relation lattice and a short basis for it can be precomputed by fixing the prime, but these computations are still subexponential and hardly scalable to larger parameters. More annoyingly, computing a powersmooth representative $\mathfrak{c}$ and evaluating the group action $\mathfrak{c} * E_0$ had conflicting requirements on B, with the optimal tradeoff also seemingly leading to a $L_p[1/3]$ subexponential-time computation [31].

3.2 SCALLOP

SCALLOP [15] modifies CSI-FiSh with the aim of improving its scalability, and particularly the class group computation bottleneck. The main change is that the order $\mathcal{O}$ is now equal to $\mathbb{Z} + f\mathcal{O}_0$, where $\mathcal{O}_0$ is the maximal order of a negative fundamental discriminant $-d_0$ of class number one, and the *conductor* f is a large prime integer. One advantage of this approach is that the class number $h(\mathcal{O}) = f - \left(\frac{-d_0}{f}\right)$ can be computed efficiently in this setting. Moreover, when h is smooth enough, then the relation lattice can be computed by solving discrete logarithms using the Pohlig–Hellman algorithm rather than index calculus. The conductor f is required to be prime for security reasons: a smooth conductor would allow inverting the group action by "walking up the volcano" [37, Theorem 5]. SCALLOP also requires a generator τ of $\mathcal{O}$ with smooth norm (for efficient computation of the group action). In the parameter selection one therefore first generates τ as a product of principal ideals of small prime norms in $\mathcal{O}_0$, until $h(\mathbb{Z}[\tau])$ is smooth enough (asymptotically $L_p[1/2]$-smooth). This computation requires subexponential time $L_p[1/2]$, but the authors of SCALLOP argue that a smaller hidden constant in the complexity makes SCALLOP more practical than CSI-FiSh [15, p. 17].

In CSI-FiSh, the Frobenius is readily available on any $\mathbb{F}_p$-rational curve and it generates the order $\mathbb{Z}[\sqrt{-p}]$. In SCALLOP one must provide an explicit representation of an endomorphism ϕ_τ corresponding to τ, or equivalently an explicit embedding $\mathcal{O} \hookrightarrow \mathrm{End}(E)$. The representation chosen aims at minimizing the size of the characteristic p while keeping all computations over $\mathbb{F}_{p^2}$. More precisely, two points P and Q on the curve are given, respectively generating the kernels of two isogenies φ_P and φ_Q such that $\phi_\tau = \widehat{\varphi}_P \varphi_Q$.

The SCALLOP paper provides an algorithm to evaluate the action of any ideal of powersmooth norm; for optimal efficiency the characteristic p is chosen such that $p - 1$ is a small multiple of the product of all primes split in $\mathcal{O}$ and smaller than a suitable bound. To evaluate the action of an arbitrary ideal, one must first compute a $L_p[1/2]$-smooth representative of that ideal using a precomputed short basis for the lattice of relations. One then computes the action of that representative, which amounts to computing $L_p[1/2]$ isogenies of small degrees and pushing torsion points trough them. So while removing the class group computation that was necessary for CSI-FiSh may result in practical improvements, evaluating the action still requires $L_p[1/2]$ subexponential time.

3.3 SCALLOP-HD

The SCALLOP-HD paper [10] revisits the SCALLOP group action using tools introduced to cryptography via the SIDH attacks [7, 27, 35]. In particular, a generator τ of $\mathcal{O}$ is now embedded into a two-power-isogeny between principally polarized abelian varieties of dimension 2 with Kani's Lemma. One first (pre-)computes two elements $\beta, \gamma \in \mathcal{O}$ such that

$$\mathrm{n}(\beta) + \mathrm{n}(\gamma) = 2^e \tag{1}$$

for some integer e. One then considers the $(2^e, 2^e)$-isogeny $F : E \times E \to E \times E$ defined by

$$F = \begin{pmatrix} \phi_{\overline{\beta}} & \phi_{\overline{\gamma}} \\ -\phi_\gamma & \phi_\beta \end{pmatrix}$$

where for any $\mu \in \mathcal{O}$ we write ϕ_μ for the corresponding endomorphism. This isogeny can be efficiently represented and evaluated at any point from its kernel, and from there one can also efficiently evaluate ϕ_β and thus ϕ_τ on any point.

The polynomial-time algorithm used to solve Equation (1) (see Section 4) is heuristic. The SCALLOP-HD authors also suggest an alternative, provable version resorting to higher-dimensional isogenies, where the sum of two norms in Equation (1) is replaced by either a sum of four norms or a sum of eight norms, both of which are easier to solve. However, they argue that only dimension-2 isogenies are currently fast enough in practice [10].

The new representation for endomorphisms greatly simplifies parameter selection in SCALLOP by removing the powersmoothness requirement on $\mathrm{n}(\tau)$. In particular, one can choose the class number polynomially smooth, so that the lattice of relations can now be computed in polynomial time by solving discrete logarithm problems in the class group.

On the other hand, the class group action is still computed by first computing a smooth norm ideal representative, then applying a class group action algorithm in the smooth case. While precomputing a short basis for the relation lattice will reduce the online costs, they can only be reduced to polynomial with an exponential time precomputation, and the optimal total cost is still subexponential.

3.4 Clapoti

The remaining subexponential costs in SCALLOP-HD come from the group action computation and its reduction to the smooth case. Clapoti [29] removes these costs with another use of Kani's lemma. The key idea for Clapoti can be seen as a generalization of what was done in SCALLOP-HD. We sketch this idea in dimension 2 for simplicity, but stress that [29] only includes concrete realizations in dimensions 4 and 8.

Let E be an $\mathcal{O}$-oriented curve and let $\mathfrak{a}$ be an $\mathcal{O}$-ideal of norm N. Let $\beta, \gamma \in \mathfrak{a}$ such that

$$\mathrm{n}_\mathfrak{a}(\beta) + \mathrm{n}_\mathfrak{a}(\gamma) = 2^e. \tag{2}$$

The ideals $\mathfrak{b} = \mathfrak{a}\overline{\beta}/N$ and $\mathfrak{c} = \mathfrak{a}\overline{\gamma}/N$ are $\mathcal{O}$-ideals in the same class as $\mathfrak{a}$. One then considers the $(2^e, 2^e)$-isogeny $F : E \times E \to E_\mathfrak{a} \times E_{\overline{\mathfrak{a}}}$ defined by

$$F = \begin{pmatrix} \phi_\mathfrak{b} & \widehat{\phi_{\overline{\mathfrak{c}}}} \\ -\phi_{\overline{\mathfrak{c}}} & \widehat{\phi_\mathfrak{b}} \end{pmatrix}$$

where $\phi_\mathfrak{b}$ and $\phi_\mathfrak{c}$ are the isogenies corresponding to $\mathfrak{b}$ and $\mathfrak{c}$. The isogeny F can be efficiently represented and evaluated at any point from its kernel, and from there one can also efficiently evaluate $\phi_\mathfrak{b}$ and $\phi_\mathfrak{c}$ on any point.

As mentioned above, the Clapoti paper only includes realizations of the above idea in dimensions 4 and 8. The sum-of-two-norms Equation (2) above is then replaced by a sum of four norms or a sum of eight norms respectively, both of which are easier to solve. While this provides a polynomial-time algorithm to evaluate the class group action in theory, working in dimension 4 or 8 renders the algorithm currently impractical in practice. On the other hand, the key obstacle to a dimension-2 realization, which we overcome in this paper for important cases, is an efficient solution to Equation (2).

4 Sum-of-two-norms equations

As discussed above, the key task for getting Clapoti to work in a given setting using only 2-dimensional isogenies is to resolve a diophantine equation of the form shown in Equation (2) in the target ideal $\mathfrak{a}$. To the best of our knowledge, Equation (2) was only introduced recently in the Clapoti paper [29], and no efficient solution was known before. However, variants of this equation, where the solutions belong to different rings, are very well-known.

When β, γ belong to $\mathbb{Z}$, the equation becomes a sum-of-two-squares equation. The equation $x^2 + y^2 = S$ has solutions if and only if all primes congruent to 3 mod 4 divide the right-hand term with even multiplicity (possibly zero). Moreover, a solution can be computed efficiently using Cornacchia's algorithm [12] when the number of distinct prime factors is small. When $S = 2^e$, we trivially have solutions $(x, y) \in \{(\pm 2^{e/2}, 0), (0, \pm 2^{e/2})\}$ when e is even, and $(x, y) \in \{(\pm 2^{(e-1)/2}, \pm 2^{(e-1)/2})\}$ when e is odd.

When β, γ belong to the ring of Gaussian integers, Equation (2) becomes a sum of four squares, and a solution always exists by Lagrange's theorem [19, Theorem 369]. Moreover, the solution can (heuristically) be efficiently computed by assigning random values to two variables and either using Cornacchia's algorithm [12] to recover values for the remaining two variables, or retrying.

An interesting generalization of Lagrange's four-squares theorem includes coefficients a, b, c, d on the squares. Ramanujan [34] proved that there are only 54 integer choices $a \leq b \leq c \leq d$ such that $aw^2 + bx^2 + cy^2 + dz^2 = n$ has a solution for all n. In fact all those tuples (a, b, c, d) are small, and the heuristic algorithm existing in the case $a = b = c = d = 1$ therefore easily generalizes to them.

Equation (1) solved in SCALLOP-HD [10] is similar to Equation (2), except that, crucially, solutions are searched for in an imaginary-quadratic order $\mathcal{O}$ instead of one of its ideals. The equation is solved as follows. Let Δ be the discriminant of $\mathcal{O}$; hence, $\mathcal{O} = \mathbb{Z}\left[\frac{\Delta+\sqrt{\Delta}}{2}\right]$ and we can write $\beta = x + \frac{\Delta+\sqrt{\Delta}}{2}$ and $\gamma = y + z\frac{\Delta+\sqrt{\Delta}}{2}$ with integer unknowns x, y, z, leading to the equation

$$(2x + \Delta)^2 + (2y + \Delta)^2 = 2^{e+2} + \Delta(z^2 + 1). \tag{3}$$

This equation can be efficiently solved by choosing random values for z until the right-hand side is a sum of two squares with the correct parity. As sums of two squares are very common, one heuristically expects to need only a few trials.

When considering β, γ in an ideal of $\mathcal{O}$ instead of $\mathcal{O}$ itself, one can still fix a basis α, N for the ideal, write $\beta = w\alpha + xN$ and $\gamma = y\alpha + zN$, and derive a quadratic diophantine equation

$$\mathrm{n}(w\alpha + xN) + \mathrm{n}(y\alpha + zN) = N2^e. \tag{4}$$

However, note that the resolution approach used for SCALLOP-HD crucially uses the fact that 1 belongs to the order, and therefore x and y only appear once in Equation (3), with small coefficients attached and no cross-term. In contrast, all quadratic terms in Equation (4) have a priori large coefficients, and the above strategy does not work.

Coming back to Lagrange's four-squares theorem, we note that some proofs involve the Hurwitz quaternions. Interestingly, our solution to Equation (5) will also use quaternions, together with the KLPT algorithm which was first introduced in [24].

5 KLaPoTi: dimension-2 Clapoti via KLPT

We now present our solution of the 2-dimensional Clapoti norm equation, which relies on KLPT.

Let $\mathcal{O}$ be a quadratic imaginary order, let $\mathfrak{a}$ be an ideal of $\mathcal{O}$ of norm N and let $\alpha \in \mathcal{O}$ be an element such that $\mathfrak{a} = (N, \alpha)$. Our goal is to solve a "sum-of-two-norms equation in the ideal" of the form

$$\mathrm{n}_\mathfrak{a}(\beta) + \mathrm{n}_\mathfrak{a}(\gamma) = 2^e$$

for some $e \in \mathbb{Z}_{>0}$, and for $\beta, \gamma \in \mathfrak{a}$.

The discussion below is restricted to the case of $\mathcal{O}$ being an order of discriminant $\Delta = -f^2 d_0$, where d_0 is a prime congruent to 3 modulo 4 such that the class number $h(-d_0)$ equals one. In particular, we will assume without loss of generality that $\mathcal{O} = \mathbb{Z}\left[\frac{1+\omega}{2}\right]$ where ω is a fixed square root of $-f^2 d_0$ in $\mathcal{O}$.

5.1 General idea

Consider the quaternion algebra $B_{-\Delta,\infty} = \mathbb{Q} + \mathbb{Q}\mathbf{i} + \mathbb{Q}\mathbf{j} + \mathbb{Q}\mathbf{ij}$ with multiplication defined by $\mathbf{i}^2 = -1$, $\mathbf{j}^2 = \Delta$, and $\mathbf{ji} = -\mathbf{ij}$. To any pair $(\beta, \gamma) \in \mathcal{O} \times \mathcal{O}$ we associate a quaternion $\beta + \mathbf{i}\gamma$ via the identification $\omega \mapsto \mathbf{j}$. Let $\mathcal{Q}$ be the quaternion order $\mathcal{O} + \mathbf{i}\mathcal{O}$ and $\mathcal{I} := \mathfrak{a} + \mathbf{i}\mathfrak{a} \subseteq \mathcal{Q}$. Note that $\mathcal{Q}$ has discriminant $|\Delta| = f^2 d_0$, and that $\mathcal{I} = \mathcal{Q}\mathfrak{a} = \mathcal{Q}N + \mathcal{Q}\alpha$ is a left $\mathcal{Q}$-ideal of norm N. Our key observation is that $\mathfrak{a} \times \mathfrak{a}$ is in bijection with $\mathcal{I}$ via the map $\mathcal{O} \times \mathcal{O} \xrightarrow{\sim} \mathcal{Q}$ from above, i.e., $\mathcal{I} = \{\beta + \mathbf{i}\gamma \mid \beta, \gamma \in \mathfrak{a}\}$, and furthermore that

$$\mathrm{n}_\mathfrak{a}(\beta) + \mathrm{n}_\mathfrak{a}(\gamma) = \mathrm{n}_\mathcal{I}(\beta + \mathbf{i}\gamma)$$

where the latter is the (reduced) quaternion norm in $B_{-\Delta,\infty}$. One can therefore apply the KLPT algorithm [24] with the Petit–Smith improvement [33] and obtain a solution of size about $|\Delta|^3 \approx f^6 d_0^3$.

In the special case where $\omega^2 = -p$, the ideal $\mathcal{I}$ corresponds to one of the two $\mathbb{F}_p$-rational isogenies of degree N; the other one is obtained by mapping $\mathbf{i} \mapsto -\mathbf{i}$.[2] More generally, the ideal $\mathcal{I}$ is distinguished among all ideals of $\mathcal{Q}$ by being one of the two ideals of norm N that is generated by an ideal of the subring $\mathcal{O}$. This leads to some simplification of the KLPT algorithm, which we detail below.

5.2 KLPT for Clapoti

As mentioned above, Equation (2) can be solved with the KLPT algorithm. However, it is worth noticing that the input provided to the algorithm for this equation is in some sense special. Indeed, the ideal $\mathcal{I} = \mathcal{Q}\mathfrak{a}$ is one of the two ideals generated by an ideal of $\mathcal{O}$, and we can write $\mathfrak{a} = \mathbb{Z}N + \mathbb{Z}\alpha$ where $\alpha = \alpha_0 + \alpha_1(1+\omega)/2$ has norm divisible by N. Writing $\beta = wN + x\alpha$ and $\gamma = yN + z\alpha$ with $w, x, y, z \in \mathbb{Z}$, the norm equation $\mathrm{n}_\mathfrak{a}(\beta) + \mathrm{n}_\mathfrak{a}(\gamma) = 2^e$ becomes

$$\mathrm{n}\big((wN + x\alpha) + \mathbf{i}(yN + z\alpha)\big) = N2^e\,,$$

which can then be written as

$$4(Nw + \alpha_0 x)^2 + 4(Ny + \alpha_0 z)^2 + |\Delta|\alpha_1^2(x^2 + z^2) = N2^{e+2}. \tag{5}$$

One can also normalize either α_0 or α_1 to 1 in this equation (but their quotient modulo N is fixed). This equation looks easier than a generic ideal norm equation solved by the KLPT algorithm, which would involve all quadratic monomials and generally distinct coefficients for all monomials.

In this section we briefly recall the various steps of the KLPT algorithm, and highlight some steps that can be simplified in our particular context. This includes a slightly more efficient way to compute a prime-ideal representative at the start of the algorithm, and a guarantee that the linear algebra step always succeeds. One may wonder whether more significant changes to the KLPT algorithm, or an altogether different algorithm, could lead to a better (shorter) solution; investigating this is left for future work.

Prime representative. This is the first step in KLPT algorithm, replacing the ideal by an ideal of prime norm in the same equivalence class. While not essential, it simplifies later steps by making non-invertible elements modulo N exponentially rare.

In KLPT we would first enumerate short elements of the *quaternion* ideal $\mathcal{I}$ until we obtain an element $\delta \in \mathcal{I}$ with prime reduced norm $\mathrm{n}_\mathcal{I}(\delta)$; then $\mathcal{I}\overline{\delta}/\mathrm{n}(\mathcal{I})$ is an equivalent $\mathcal{Q}$-ideal of prime norm $\mathrm{n}_\mathcal{I}(\delta)$, so we may perform the rest of the KLPT algorithm with $\mathcal{I}\overline{\delta}/\mathrm{n}(\mathcal{I})$ in place of $\mathcal{I}$. For a random ideal of $\mathcal{Q} = \mathcal{O} + \mathbf{i}\mathcal{O}$ the reduced basis elements all have norm roughly $|\Delta|^{1/2}$, and the prime norm representative is (on average) only bigger than that by a logarithmic factor.

[2] In general, $p \neq -\Delta$ hence this is *not* the same quaternion algebra $B_{p,\infty}$ in which endomorphism rings of supersingular elliptic curves over $\overline{\mathbb{F}_p}$ embed.

Here we have the following alternative option: Enumerate short elements of the *quadratic* ideal $\mathfrak{a}$ until an element $\delta \in \mathfrak{a}$ with prime reduced norm $\mathrm{n}_{\mathfrak{a}}(\delta)$ is found; then $\mathfrak{a}' := \mathfrak{a}\overline{\delta}/\mathrm{n}(\mathfrak{a})$ is an equivalent $\mathcal{O}$-ideal of prime norm $\mathrm{n}_{\mathfrak{a}}(\delta)$. Finally, compute the corresponding quaternion ideal $\mathcal{Q}\mathfrak{a}'$. The shortest representative of any ideal class of $\mathcal{O}$ has norm bounded by $|\Delta|^{1/2}$ since every class of binary quadratic forms contains a reduced representative, and the shortest prime representative is expected to have norm only slightly larger than that (heuristically, by a single logarithmic factor).

Overall, the two approaches are expected to produce ideals of similarly-sized norms. Using the second method is slightly more efficient as the manipulations involve smaller lattices, though the benefit will be negligible in the context of the overall computation.

Represent integer. The next step in KLPT algorithm is to find an element γ_1 in $\mathcal{Q}$ with norm $N2^{e_1}$. This step amounts to finding elements ρ, ν in the subring $R := \mathbb{Z}[\mathbf{i}]$ of $\mathcal{Q}$ such that

$$\mathrm{n}(\rho) + \Delta\mathrm{n}(\nu) = N2^{e_1}$$

and letting $\gamma_1 := \rho + \mathbf{j}\nu$. (Here the norms are from R.) In KLPT one generates ν randomly and small until $N2^{e_1} - \Delta\mathrm{n}(\nu)$ is "Cornacchia-friendly", i.e., such that Cornacchia's algorithm will find a solution quickly.

Anticipating the next step, we choose $\nu \neq 0$, which is automatically satisfied if "Cornacchia-friendly" is interpreted as "a prime congruent to 1 modulo 4".

Reduction to a simplified lifting step. In this step, we search for $\mu_0 \in R\mathbf{j}$ such that $\gamma_1\mu_0 \in \mathcal{I}$. This will ensure that $\mathcal{J} = \mathcal{I}\overline{\mu_0\gamma_1}/N$ is in the same class as $\mathcal{I}$, while replacing μ_0 with a suitably smooth lift (in the next step) will ensure that the norm of $\mathcal{J}$ is as required.

This is just a small linear algebra system, posing no computational challenge once the existence of a solution is guaranteed. The KLPT paper [24, Section 4.3] argues about existence based on the orbits of the action of $R\mathbf{j}/N\mathcal{Q}$ over the ideals of norm N: When N is inert there is a single orbit, hence the system always has a solution. When N is split, there is a large orbit of size $N-1$ and a small orbit of size 2, and a solution exists whenever $\mathcal{I}$ and $\mathcal{Q}N + \mathcal{Q}\gamma_1$ are in the same orbit. The two elements in the small orbit are of the form $\mathcal{Q}N + \mathcal{Q}\gamma_1$ with $\gamma_1 \in R$.

In our case, $\mathcal{I}$ is in the large orbit because it is generated by elements of $\mathcal{O}$. Forcing $\nu \neq 0$ in the represent-integer algorithm will ensure $\gamma_1 \notin R$, so that this step always succeeds.

Lifting step/strong approximation. Given $\mu_0 \in R\mathbf{j}/N\mathcal{Q}$, this step aims to compute a lift in $\mathcal{Q}$ with suitable norm. The goal is to compute $\mu = \delta\mu_0 + N\mu_1$ of form 2^{e_2}. The element $\gamma_1\mu \in \mathcal{I}$ is then of norm $N2^{e_1+e_2}$. Consequently, $\mathcal{I}\overline{\gamma_1\mu_0}/N$ has norm $2^{e_1+e_2} = 2^e$.

As the input μ_0 not only depends on $\mathcal{I}$ but also on γ_1, it is not clear how the special form of $\mathcal{I}$ can help to improve this step. To reduce the overall output size we use the Petit–Smith's lifting algorithm here [33].

Summary of KLaPoTi. We summarize the use of the KLPT algorithm for Clapoti in Algorithm 1.

Algorithm 1: KLPT for Clapoti.

Input: An integral ideal $\mathfrak{a}$ of norm N of a quadratic imaginary order $\mathcal{O}$ and a large enough target norm 2^e.

Output: A solution to the Clapoti equation: Elements $\beta, \gamma \in \mathfrak{a}$ such that $\gcd(\mathrm{n}_\mathfrak{a}(\beta), \mathrm{n}_\mathfrak{a}(\gamma)) = 1$ and $\mathrm{n}_\mathfrak{a}(\beta) + \mathrm{n}_\mathfrak{a}(\gamma) = 2^e$.

1: Let $\mathcal{I} = \mathfrak{a} + \mathbf{i}\mathfrak{a}$.
2: Replace $\mathcal{I}$ by an equivalent ideal of prime norm.
3: Find $\gamma_1 \in \mathcal{Q}$ with norm $N2^{e_1}$, where $e_1 < e$, and let $e_2 := e - e_1$.
4: Find $\mu_0 \in R\mathbf{j}$ such that $\gamma_1\mu_0 \in \mathcal{I}$ ▷ $\mathcal{J} = \mathcal{I}\overline{\mu_0\gamma_1}/N$ is equivalent to $\mathcal{I}$.
5: Find $\mu = \delta\mu_0 + N\mu_1$ of norm 2^{e_2} ▷ $\gamma_1\mu \in \mathcal{I}$ is of norm $N2^e$.
6: Extract β, γ from the quaternion element $\beta + \mathbf{i}\gamma := \gamma_1\mu$.
7: If $\gcd(\mathrm{n}_\mathfrak{a}(\beta), \mathrm{n}_\mathfrak{a}(\gamma)) \neq 1$, retry with new randomness.
8: **Return** (β, γ)

6 The SCALLOP2D group action

In this section, we apply KLaPoTi to construct a new variant of SCALLOP where both the orientation and the group action are represented and computed using dimension-2 isogenies.

We start by describing the group action and we then briefly discuss protocols that can be built on top of it. Following that, we present two optimizations of the group action in particular protocol contexts.

6.1 The group action

Representing oriented curves. An $\mathcal{O}$-oriented curve can be described by a j-invariant which can be expanded to a curve E using a canonical method, and an endomorphism $\tau \in \mathrm{End}(E)$. The endomorphism itself is represented by the two points $\big(\tau(P), \tau(Q)\big)$ where P, Q are a deterministically generated basis of the 2^e-torsion on E. Such a representation requires 3 elements in $\mathbb{F}_{p^2}$, plus a constant number of bits for sign disambiguation, amounting to roughly $6 \log p$ bits in total.

For efficiency, the points P, Q may be encoded as part of the representation instead of being reconstructed on the fly; this way, an $\mathcal{O}$-oriented curve is stored using $10 \log p$ bits in total. This option offers a size vs. computation trade-off.

Group action evaluation. There is a transitive action of the ideal class group of $\mathcal{O}$ on the set of primitively $\mathcal{O}$-oriented elliptic curves [11, 28]. We now show how to evaluate this group action for our SCALLOP variant.

The group action evaluation consists of two distinct phases, one that only depends on $\mathcal{O}$, followed by one that depends on the particular $\mathcal{O}$-oriented curve in use. In many application scenarios, the first phase can be computed during key generation, and only the second phase needs to be performed online; in particular, this is the case for a CSIDH-like key exchange.

Our main contribution compared to previous works lies in the first phase. Let $\mathfrak{a}$ be an $\mathcal{O}$-ideal. Using the approach described in Section 5, we compute $\beta, \gamma \in \mathfrak{a}$ such that

$$\mathrm{n}_\mathfrak{a}(\beta) + \mathrm{n}_\mathfrak{a}(\gamma) = 2^e$$

with both summands odd and coprime. We then compute the ideals $\mathfrak{b} = \mathfrak{a}\overline{\beta}/N$ and $\mathfrak{c} = \mathfrak{a}\overline{\gamma}/N$, both of which are integral $\mathcal{O}$-ideals lying in the same class as $\mathfrak{a}$, as well as the principal ideal $\mathfrak{d} = \mathfrak{b}\overline{\mathfrak{c}}$, and compute $u, v \in \mathbb{Z}$ such that $\mathfrak{d} = (u + v\tau)\mathcal{O}$. This completes the first phase of the algorithm.

The second phase follows the Clapoti framework [29]. We detail the steps here for completeness. We are given an $\mathcal{O}$-oriented curve, represented by a curve E and four points $\big(P, Q, \tau(P), \tau(Q)\big)$, as described above. As in Clapoti, we consider the $(2^e, 2^e)$-isogeny $\Phi : E \times E \to E_\mathfrak{a} \times E_{\overline{\mathfrak{a}}}$ coming from the following diagram:

$$\begin{array}{ccc} E & \xrightarrow{\phi_\mathfrak{b}} & E_\mathfrak{a} \\ {\scriptstyle \phi_{\overline{\mathfrak{c}}}}\downarrow & & \downarrow{\scriptstyle \phi_{\overline{\mathfrak{c}}}} \\ E_{\overline{\mathfrak{a}}} & \xrightarrow[\phi_\mathfrak{b}]{} & E \end{array}$$

The isogeny Φ is thus defined by

$$\Phi = \begin{pmatrix} \phi_\mathfrak{b} & \widehat{\phi}_{\overline{\mathfrak{c}}} \\ -\phi_{\overline{\mathfrak{c}}} & \widehat{\phi}_\mathfrak{b} \end{pmatrix} . \tag{6}$$

Its kernel is

$$\ker \Phi = \left\langle \big([\mathrm{n}(\mathfrak{b})]R, \widehat{\phi}_\mathfrak{c}\phi_\mathfrak{b}(R)\big) \ : \ R \in \{P, Q\} \right\rangle .$$

The kernel be computed by finding $u, v \in \mathbb{Z}$ such that $u + v\tau = \widehat{\phi}_\mathfrak{c}\phi_\mathfrak{b}$, which is easy since $\widehat{\phi}_\mathfrak{c}\phi_\mathfrak{b}\mathcal{O} = \mathfrak{b}\overline{\mathfrak{c}}$, and thus evaluating $\widehat{\phi}_\mathfrak{c}\phi_\mathfrak{b}(R)$ as $[u]R + [v]\tau(R)$ for $R \in \{P, Q\}$ using the images of P, Q given as part of the orientation. From $\ker \Phi$ we compute Φ using theta coordinates, using the formulas of [14].

The theta isogeny formulas yield the two elliptic factors of the image of Φ, but it is a priori unknown which one is which, so we are left with a set of two curves $\{E_1, E_2\} = \{E_\mathfrak{a}, E_{\overline{\mathfrak{a}}}\}$ and the corresponding projection maps

$$\pi_i : \quad E \times E \xrightarrow{\Phi} E_\mathfrak{a} \times E_{\overline{\mathfrak{a}}} \xrightarrow{\mathrm{proj}} E_i \, .$$

At this point, it remains to identify the correct component $E_\mathfrak{a}$, and to compute the data defining the orientation on that curve.

Differentiation between $E_\mathfrak{a}$ and $E_{\overline{\mathfrak{a}}}$ can be done using pairings, as described in [29]. Let ϕ_i be the elliptic curve isogeny $E \to E_i$ defined by $R \mapsto \pi_i((R, 0))$.

Indeed, from Equation (6) we see that $\phi_i(R)$ must for all points $R \in E$ be either $\phi_{\mathfrak{b}}(R)$ or $-\phi_{\bar{\mathfrak{c}}}(R)$, depending on whether $E_i = E_{\mathfrak{a}}$ or $E_i = E_{\bar{\mathfrak{a}}}$. To distinguish the two cases, one may thus exploit the fact that the Weil pairing is compatible with isogenies: We have $e_t(\phi_i(P), \phi_i(Q)) = e_t(P,Q)^{\deg \phi_i}$ for all points $P, Q \in E[t]$. Using a t-torsion basis for some small t such that $\mathrm{n}(\mathfrak{b}) \not\equiv \mathrm{n}(\mathfrak{c}) \pmod{t}$, we may thus identify whether $\phi_i = \phi_{\mathfrak{b}}$ or $\phi_i = -\phi_{\bar{\mathfrak{c}}}$. In practice, it is convenient to choose t as a (small) power of two, since (see below) points of two-power order have to be pushed through Φ anyway in order to propagate the orientation to the new curve $E_{\mathfrak{a}}$.

To compute an embedding of τ in $E_{\mathfrak{a}}$, we proceed as in SCALLOP-HD [10]. Since $\phi_{\mathfrak{a}}$ and $\phi_{\mathfrak{b}}$ both commute with $\mathcal{O}$ by definition of the group action, the four-tuple

$$\Big(\phi_{\mathfrak{b}}(P),\ \phi_{\mathfrak{b}}(Q),\ \phi_{\mathfrak{b}}\big(\tau(P)\big),\ \phi_{\mathfrak{b}}\big(\tau(Q)\big)\Big) \tag{7}$$

is a valid representation of the orientation by $\mathcal{O}$ on $E_{\mathfrak{a}}$ induced by $\phi_{\mathfrak{a}}$, and these four points can be efficiently computed using the 2-dimensional representation Φ of $\phi_{\mathfrak{b}}$ once the component of the codomain of Φ which corresponds to $E_{\mathfrak{a}}$ has been identified. Converting this representation of the orientation using four points into the more compact representation using a deterministic basis and two points is straightforward by computing discrete logarithms (possibly using pairings as an optimizations) in a group of two-power order and some linear algebra.

It should also be noted that standardizing the representation of the orientation is in fact crucial for security: Revealing $\phi_{\mathfrak{b}}(P)$ and $\phi_{\mathfrak{b}}(Q)$ directly would provide nontrivial information about the degree of $\phi_{\mathfrak{b}}$ to an attacker, which could then possibly be exploited in a variant of the SIDH attack [7, 27, 35]; see also [16].

Sampling random ideals. For the sake of completeness, we describe a simple algorithm to sample random secret ideals $\mathfrak{a} \subseteq \mathcal{O}$ in Algorithm 3. We note that a significantly faster method for the same task has recently been proposed in [9].

System parameter generation. Like in SCALLOP, the orientation is chosen as $\mathcal{O} = \mathbb{Z} + f\mathcal{O}_0$ where f is a large prime. We also assume $p \equiv 3 \pmod 4$ for simplicity. By using the Clapoti group action evaluation, we have no smoothness requirement on the class number $h(\mathcal{O})$.

Finally, we compute an initial $\mathcal{O}$-oriented curve using Eriksen's SageMath implementation[3] of [2, Algorithm 3]: Starting from a "special" starting curve E_0 with known endomorphism ring and an endomorphism τ/f, we compute the images of a 2^e-torsion basis of a degree-f isogeny $\phi_I \colon E_0 \to E$ given by an ideal I as in Lemma 2, and deduce the action of $\phi_I \circ (\tau/f) \circ \widehat{\phi_I}$ on the 2^e-torsion. This information encodes an orientation by τ on E as a "2dim representation", following the terminology of [10].

[3] https://github.com/Jonathke/deuring-2D

Algorithm 2: Group action.

Input: An integral ideal $\mathfrak{a}$ of an imaginary-quadratic order $\mathcal{O}$ of norm N and a large enough target norm 2^e; an $\mathcal{O}$-oriented curve (E, τ) where τ is represented by the images $\tau(P), \tau(Q)$ of a 2^e-torsion basis (P, Q) on E.

Output: $\mathcal{O}$-oriented curve $(E_\mathfrak{a}, \tau')$.

Precomputation part (only dependent on $\mathfrak{a}$ but not E)

1: Compute a pair (β, γ) of elements in $\mathcal{O}$ such that $\gcd(\mathrm{n}_\mathfrak{a}(\beta), \mathrm{n}_\mathfrak{a}(\gamma)) = 1$ and $\mathrm{n}_\mathfrak{a}(\beta) + \mathrm{n}_\mathfrak{a}(\gamma) = 2^e$ using KLPT for Clapoti.
2: Compute the two ideals equivalent to $\mathfrak{a}$: $\mathfrak{b} = \mathfrak{a}\overline{\beta}/N$ and $\mathfrak{c} = \mathfrak{a}\overline{\gamma}/N$.
3: Compute the principal ideal $\mathfrak{d} = \mathfrak{b}\overline{\mathfrak{c}}$ and find a generator $u + v\tau \in \mathcal{O}$.

Online part (dependent on both $\mathfrak{a}$ and E)

4: Compute the kernel of $\Phi : E \times E \to E_\mathfrak{a} \times E_{\overline{\mathfrak{a}}}$, namely

$$\left\langle \left([\mathrm{n}(\mathfrak{b})]R, (u + v\tau)(R)\right) \ : \ R \in \{P, Q\} \right\rangle.$$

5: Compute Φ using $\ker \Phi$ and theta coordinates.
6: Differentiate between $E_\mathfrak{a}$ and $E_{\overline{\mathfrak{a}}}$ using the Weil pairing.
7: Push the points $P, Q, \tau(P), \tau(Q)$ through $\phi_\mathfrak{a}$ to obtain a representation of the new orientation $\tau' \in \mathrm{End}(E_\mathfrak{a})$ on $E_\mathfrak{a}$.
8: **Return** $(E_\mathfrak{a}, \tau')$.

Algorithm 3: Sampling of the secret ideal $\mathfrak{a}$.

Input: Imaginary-quadratic order $\mathcal{O} = \mathbb{Z}[\tau]$ of discriminant Δ.
Output: Random ideal $\mathfrak{a}$ of $\mathcal{O}$ whose class is close to uniform.

1: Let $\chi \in \mathbb{Z}[X]$ be the minimal polynomial of τ.
2: Find a prime $\ell \gg |\Delta|$ such that χ has a root λ modulo ℓ.
3: **Return** $\mathfrak{a} = (\ell, \tau - \lambda)\mathcal{O}$.

Security considerations. Let λ be the security parameter, i.e., we want the best attacks to cost at least 2^λ operations.

The endomorphism ring computation problem must be hard as otherwise the group action can be efficiently inverted [37, Section 6]; this requires $\log p \geq 2\lambda$ to protect against classical cycle-finding attacks [18], and $\log p \geq 4\lambda$ to protect against quantum versions of this attack using Grover's algorithm.

As in previous SCALLOP variants, we also require $\mathcal{O}$ to have class number h bounded by $\log h \geq 2\lambda$ bits to prevent meet-in-the-middle search attacks against the vectorization problem. This implies $\log \Delta \geq 4\lambda$ and thus $\log f \geq 2\lambda$.

Quantum security for the vectorization problem is much harder to assess. Asymptotically the best attacks are variants of Kuperberg's algorithm [26, 32], which have $L_h[1/2]$ subexponential time and gate complexity, forcing $\log \Delta$ to scale quadratically with λ. It turns out that evaluating the group action contributes to an important factor in those costs [3]. One can roughly infer the quantum security level for our group action from existing analysis on CSIDH [32, 6], provided a cost estimate for a quantum circuit evaluating the group action.

We leave this to further work, but note that the original CSIDH-512 parameters may be considered too small depending on the model [32, 6]; see also [30, § 11.4].

Efficiency considerations. Recent high-dimensional techniques like Clapoti have largely lifted the smoothness and powersmoothness requirements that previously were the marks of KLPT-like techniques. In SCALLOP and previous variants, these requirements also came from the relation lattice.

Paradoxically, in this paper we obtain a practical improvement to Clapoti by relying on KLPT and re-introducing smoothness requirements.

More precisely, the KLPT algorithm [24], in its Petit–Smith variant [33], gives output size $|\Delta|^2 N^2$ where N is the norm of the target ideal. We a priori expect $N \approx |\Delta|^{1/2}$, resulting in an output size of about $|\Delta|^3$. Torsion points of order $2^e \approx |\Delta|^3$ will need to be efficiently represented. We choose $p = 2^e c - 1$ for a small cofactor c and work with the component of the isogeny graph containing curves defined over $\mathbb{F}_p$, so that the entire 2^e-torsion is $\mathbb{F}_{p^2}$-rational. Together with the security considerations above, this means one needs to take $\log p \approx 12\lambda$ for λ bits of security against classical attacks.

Comparison with other isogeny group actions. Compared to CSIDH, SCALLOP and SCALLOP-HD, our group action is fully scalable, requiring no superpolynomial computation for either the class group, a relation lattice, a short basis of that lattice, or parameter search with smoothness requirements. Moreover, the use of 2-dimensional isogenies, as opposed to 4- or 8-dimensional isogenies in Clapoti, makes it reasonably efficient as shown by our preliminary Rust implementation results (Section 7).

Public keys for our group action SCALLOP2D will generally be a factor of 3 to 5 times larger than for CSIDH. Comparisons with SCALLOP and SCALLOP-HD are exacerbated by the fact that they depend on various parameter choices: There is a tradeoff between key sizes and efficiency in those schemes. Public keys in protocols based on all these group actions typically comprise of one $\mathcal{O}$-oriented curve. This requires $2\log p$ bits for CSI-FiSh, $6\log p$ bits for SCALLOP, and $10\log p$ bits for SCALLOP-HD and SCALLOP2D ($6\log p$ with compressed representations). In SCALLOP2D, this leads to choosing 12λ-bit characteristic p as explained above. For CSI-FiSh, SCALLOP and SCALLOP-HD, the link between p and λ is less clear: keeping all computations over $\mathbb{F}_{p^2}$ (as suggested in those works) will require p to grow superpolynomially in λ, but for large parameters one may also drop the requirement and prefer to incur the practical overhead costs of working with moderate degree extension fields instead (on top of other superoplynomial factors in the cost for those schemes).

For the classical 128-bit security level, total public key sizes for SCALLOP2D can therefore be about 1152 bytes, as opposed to 128, 469, and 852 bytes for CSI-FiSh, SCALLOP, and SCALLOP-HD.

6.2 Group-action-based cryptographic protocols

Starting from an effective group action as above, a whole range of cryptographic protocols can be built, including CSI-FiSh-like signatures [4], CSIDH-like key exchange [8], and many more [1]. Since instantiating these general constructions is straightforward from our description of the group action and the literature, we omit their details here, and only focus on relevant optimizations.

6.3 Protocol-specific optimizations

In this section we provide two optimizations to the SCALLOP2D group action which are applicable in some, but a priori not all, group-action-based isogeny protocols. The first optimization allows to avoid pairing computation, and applies to a CSIDH-type non interactive key exchange. The second optimization replaces canonical representations of oriented curves by carefully randomized ones, and applies to a CSI-FiSh-like signature protocol. These optimizations might also apply to other group-action-based isogeny protocols.

Optimization 1: avoiding pairing computation. In the final step of the $(2,2)$-isogeny walk, one must split a principally polarized abelian variety as a product of two elliptic curves $E_{\mathfrak{a}} \times E_{\overline{\mathfrak{a}}}$, and in particular identify its projection onto $E_{\mathfrak{a}}$ to evaluate $\phi_{\mathfrak{b}}$ on given points. As described above, this can be done using two pairing computations. While efficient pairing algorithms exist, this certainly adds to computation time and code complexity.

In this section, we observe that instead of identifying the correct component, one can often simply work with both.

Following [37], we define a twisting operation on $\mathcal{O}$-oriented curves by

$$(E,\tau)^t = (E^{(p)}, \pi \circ \overline{\tau} \circ \pi^{-1})$$

where π is the p-power Frobenius isogeny $E \to E^{(p)}$. Then, for any $\mathfrak{a} \in \mathcal{O}$,

$$\overline{\mathfrak{a}} * (E,\tau)^t = \big(\mathfrak{a} * (E,\tau)\big)^t,$$

see [37, Lemma 7]. Moreover, we observe that when E is defined over $\mathbb{F}_p$ and τ anticommutes with the p-power Frobenius endomorphism on E, then the twisting operation leaves the oriented curve (E,τ) invariant:

Lemma 1. *Let E be a supersingular elliptic curve over $\mathbb{F}_p$ and let $\tau \in End(E)$ such that $\tau\pi = -\pi\tau$, where π is the p-power Frobenius endomorphism on E. Then $(E,\tau)^t = (E,\tau)$.*

Proof. Clearly $E^{(p)} = E$ since $E/\mathbb{F}_p$. For the second component, first notice that τ must have trace zero: On the one hand $\operatorname{tr}(\tau)\pi = \pi \operatorname{tr}(\tau)$ since $\operatorname{tr}(\tau) \in \mathbb{Z}$, but on the other hand $\operatorname{tr}(\tau)\pi = (\tau+\overline{\tau})\pi = -\pi(\tau+\overline{\tau}) = -\pi \operatorname{tr}(\tau)$ since $\tau\pi = -\pi\tau$ and hence also $\overline{\pi}\tau = -\pi\overline{\tau}$. Thus $\overline{\tau} = -\tau$, and it follows that

$$\pi\overline{\tau} = -\pi\tau = \tau\pi\,.$$

This proves the claim $\pi\overline{\tau}\pi^{-1} = \tau$. □

We also observe that one can efficiently compute an $\mathcal{O}$-oriented curve defined over $\mathbb{F}_p$ for the case $\mathcal{O} \cong \mathbb{Z}[f\vartheta]$ where $\mathbb{Z}[\vartheta]$ has small discriminant:

Lemma 2. *Fix $p \geq 5$ and let E_0 be a supersingular elliptic curve defined over $\mathbb{F}_p$ whose endomorphism ring $\mathcal{Q}_0$ contains an endomorphism $\vartheta \in \mathcal{Q}_0$ that anti-commutes with the p-power Frobenius endomorphism π of E_0. Given a prime $f \equiv 3 \pmod 4$ not dividing $\operatorname{disc}(\mathbb{Z}[\vartheta])$ modulo which $-p$ is a square, let $\lambda \in \mathbb{Z}$ satisfy $\lambda^2 \equiv -p \pmod f$, and let $I := \mathcal{Q}_0 f + \mathcal{Q}_0(\pi - \lambda)$. Then I corresponds to an $\mathbb{F}_p$-rational isogeny $\phi_I \colon E_0 \to E$. The curve $E/\mathbb{F}_p$ together with a concrete embedding of $\mathcal{O} \cong \mathbb{Z}[f\vartheta]$ in $\mathrm{End}(E)$ can be computed in polynomial time using KLPT-based [25] or higher-dimensional techniques [2, Algorithm 3].*

Proof. The claim that ϕ_I is an $\mathbb{F}_p$-rational isogeny follows immediately from the fact that I is generated by endomorphisms contained in the subring $\mathbb{Z}[\pi]$.

Now define $\tau := \phi_I \circ \vartheta \circ \widehat{\phi}_I$. Clearly $\mathcal{O} := \mathbb{Z}[\tau]$ is isomorphic to $\mathbb{Z}[f\vartheta]$: Let $\mu = X^2 - tX + n \in \mathbb{Z}[X]$ be the minimal polynomial of ϑ; hence, the minimal polynomial of $f\vartheta$ is $\mu' = X^2 - ftX + f^2 n$. On the other hand,

$$\tau^2 = f\phi_I \vartheta^2 \widehat{\phi}_I = f\phi_I(t\vartheta - n)\widehat{\phi}_I = ft\phi_I \vartheta \widehat{\phi}_I - f^2 n = ft\tau - f^2 n\,,$$

which shows $\mu'(\tau) = 0$. Thus $\tau \mapsto f\vartheta$ defines a ring isomorphism $\mathbb{Z}[\tau] \to \mathbb{Z}[f\vartheta]$.

To prove that the orientation of E by τ is primitive, it suffices to show that $\ker \phi_I = E_0[I]$ does not form an eigenspace of ϑ: Since $E_0[I] = (\widehat{\pi} - \lambda)(E_0[f])$ and $\widehat{\pi} = -\pi$, we get

$$\begin{aligned}\vartheta(E_0[I]) &= \big(\vartheta(\widehat{\pi} - \lambda)\big)(E_0[f]) = \big(\vartheta(\pi + \lambda)\big)(E_0[f]) \\ &= \big((\pi - \lambda)\vartheta\big)(E_0[f]) = (\pi - \lambda)(E_0[f]) = E_0[\overline{I}]\,.\end{aligned}$$

Any nonzero point $P \in E_0[I] \cap E_0[\overline{I}]$ must simultaneously satisfy $\pi(P) = [\lambda]P$ and $\pi(P) = [-\lambda]P$, hence $\lambda^2 \equiv -1 \pmod f$, which would contradict the assumption that $f \equiv 3 \pmod 4$. Therefore $\vartheta(E_0[I]) \cap E_0[I] = \{\infty\}$. □

We are now ready to sketch how our optimization works, in the particular case of a CSIDH-like key exchange protocol. In a straightforward application of the protocol, public keys would be of the form $\mathfrak{a} * E$ and $\mathfrak{b} * E$, and the shared secret would be equal to $\mathfrak{ab} * E$, which is then typically fed into a key derivation function.

Under the above restrictions, we can use public keys of the same shape; however instead of computing the correct component $\mathfrak{a} * E$, we just compute a random component $\mathfrak{a}^{\epsilon_a} * E$ for $\epsilon_a \in \{-1, 1\}$ (and similarly for the other party). No pairing computation is needed here because the elliptic curve component is chosen randomly.

As for the shared secret, given $\mathfrak{b}^{\epsilon_b} * E$, one applies the action of $\mathfrak{a}$ as above, up to the points where the two components $\mathfrak{a}\mathfrak{b}^{\epsilon_b} * E$ and $\mathfrak{a}^{-1}\mathfrak{b}^{\epsilon_b} * E$ are obtained. Note that no pairing computation is applied, so although we have both curves we do not know which one is which. Then one simply applies twists to both curves to obtain four curves $\mathfrak{a}^{\epsilon_a}\mathfrak{b}^{\epsilon_b} * E$ for $\epsilon_a, \epsilon_b \in \{-1, 1\}$. Finally, the four corresponding j-invariants are ordered in a canonical way, and a key derivation function is applied to the resulting tuple.

Remark. Computing an isogeny between two $\mathcal{O}$-oriented curves does not seem to become easier when the starting curve is one of the two "special" $\mathcal{O}$-oriented curves that are also defined over $\mathbb{F}_p$, much like starting from the curve with j-invariant 1728 does not make the isogeny problem easier in CSIDH. Indeed, assuming there exists an algorithm that solves the isogeny problem from one special curve, one can call this algorithm twice to solve the isogeny problem between any two $\mathcal{O}$-oriented curves.

Optimization 2: avoiding canonical representations. Within the context of our preliminary SageMath implementation we observed that a significant fraction of the total time was spent on computing a *canonical* representation of the information needed to represent oriented curves. This procedure has been applied since [15], both to allow the generation of unique shared secrets, and to avoid leaking any (torsion-point) information about previously computed group actions. (The leakage of information is not a mere theoretical possibility, as explained in Section 6.) While seemingly innocent, this computation actually contributes to significant computation costs.

We observe that in some contexts, and in particular when generating public keys, or when computing challenge curves inside a CSI-FiSh-style signature algorithms, a unique representation is not needed, and the canonicalization can be replaced by a more efficient randomization procedure. More precisely, after computing the tuple from Equation (7), which is a valid representation but leaks information on previously computed group actions, we generate a random invertible matrix $M \in \mathrm{GL}_2(\mathbb{Z}/2^e)$ and construct the new representation by applying the matrix M to the first and second half of the tuple separately. Note that the randomization only involves a couple of "bi-scalar multiplications": elliptic-curve point computations of the form $[a]P + [b]Q$.

7 Implementation

To demonstrate the practical feasibility of implementing and running our new cryptographic group action, we provide a semi-optimized implementation of SCALLOP2D. However, we emphasize once more that the main purpose of this work is not to achieve practical speed improvements over existing group actions, but to achieve *polynomial-time* asymptotic scaling while retaining the concrete instantiability and practicality of the construction.

We present implementations of our algorithm in SageMath and Rust. The Rust implementation runs about 70 times faster than the equivalent SageMath implementation for a parameter set with $\log_2 |\Delta| \approx 512$.

7.1 SageMath implementation

Our SageMath implementation of SCALLOP2D is loosely based on the SCALLOP-HD proof-of-concept implementation[4], which in turn relies on the imple-

[4] https://github.com/isogeny-scallophd/scallophd

mentation of the Deuring correspondence from [25] and the implementation of $(2^e, 2^e)$-isogenies in the theta model from [14].

In particular, the generation of the starting curve and general algorithmic wrappers for "2dim representations" of 1-dimensional isogenies are borrowed from there. The KLPT implementation for our proof-of-concept implementation of SCALLOP2D was written from scratch in SageMath.

Timings are not representative of actual performance potential due to various overhead costs affecting the performance of SageMath in general, but to give a rough ballpark figure, we mention that the SageMath implementation for a 512-bit discriminant Δ takes about three minutes on a standard consumer laptop. Given our plans to proceed with a Rust implementation, we made no serious attempt at optimizing the SageMath implementation further.

Code. Our SageMath implementation is available here:
https://github.com/isogeny-klapoti/klapoti-sage

7.2 Rust implementation

We designed our Rust implementation to be reusable and to offer the functionality needed for implementing isogeny-based cryptographic protocols.

In recent years, the world of isogeny-based cryptography gravitated from C and C++ to Rust. Interestingly, the implementation of building blocks for dimension-2 isogenies was provided in [14], but there is still a lack of Rust building blocks for dimension-1 isogenies. Our library aims to at least partially close that gap. We provide the functionality for big integer matrices, lattices, quaternion algebras, and number fields which are required for dimension-1 isogenies, but not for dimension-2. Additionally, we provide the first Rust implementation of the Cornacchia and KLPT algorithms.

Our work is based on the following libraries:

- The `two-isogenies` Rust implementation of $(2^e, 2^e)$-isogenies from [14].
- `two-isogenies` in turn is based on Pornin's `crrl` library[5] for finite field arithmetic and elliptic curves.
- The quaternion parts are heavily influenced by the C implementation of SQIsign submitted to NIST[6].

The Rust implementation takes about 2.5 wall-clock seconds on standard hardware for a single group-action evaluation with a 512-bit discriminant Δ. (The equivalent SageMath implementation takes approximately three minutes.) We report the timings of the Rust implementation in Tables 1 and 2.

While there are other challenges remaining to be addressed in the future (including a constant-time implementation), we believe that our work provides an important building block in the isogeny-based cryptography Rust ecosystem.

Code. Our Rust implementation is available here:
https://github.com/isogeny-klapoti/klapoti-rust

[5] https://github.com/pornin/crrl
[6] https://github.com/SQISign/the-sqisign

Table 1: Timing results for a single group-action evaluation of our Rust implementation of the SCALLOP2D group action (mean of 100 measurements). Times are given in wall-clock time on an Apple M1 MacBook Pro clocked at 3.2 GHz. Generating parameter sets for larger security levels is work in progress.

$\approx \log_2 \lvert\Delta\rvert$	64	128	512
KLPT	3.58 ms	37.89 ms	0.43 s
Isogenies	3.56 ms	12.18 ms	2.06 s

Table 2: Mean number of CPU cycles for a single group-action evaluation of our Rust implementation of the SCALLOP2D group action on Apple M1 MacBook Pro clocked at 3.2 GHz.

$\approx \log_2 \lvert\Delta\rvert$	64	128	512
KLPT	1.14e7	1.21e8	1.38e9
Isogenies	1.14e7	3.90e7	6.59e9

References

[1] Navid Alamati, Luca De Feo, Hart Montgomery, and Sikhar Patranabis. "Cryptographic Group Actions and Applications". In: *ASIACRYPT (2)*. Vol. 12492. LNCS. Springer, 2020, pp. 411–439. URL: https://ia.cr/2020/1188.

[2] Andrea Basso, Luca De Feo, Pierrick Dartois, Antonin Leroux, Luciano Maino, Giacomo Pope, Damien Robert, and Benjamin Wesolowski. *SQIsign2D-West: The Fast, the Small, and the Safer*. To appear at Asiacrypt 2024. 2024. URL: https://ia.cr/2024/760.

[3] Daniel J. Bernstein, Tanja Lange, Chloe Martindale, and Lorenz Panny. "Quantum Circuits for the CSIDH: Optimizing Quantum Evaluation of Isogenies". In: *EUROCRYPT (2)*. Vol. 11477. LNCS. Springer, 2019, pp. 409–441. URL: https://ia.cr/2018/1059.

[4] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. "CSI-FiSh: Efficient Isogeny based Signatures through Class Group Computations". In: *ASIACRYPT (1)*. Vol. 11921. LNCS. Springer, 2019, pp. 227–247. URL: https://ia.cr/2019/498.

[5] Jean-François Biasse. "Improvements in the computation of ideal class groups of imaginary quadratic number fields". In: *Advances in Mathematics of Communication* 4.2 (2010), pp. 141–154.

[6] Xavier Bonnetain and André Schrottenloher. "Quantum Security Analysis of CSIDH". In: *EUROCRYPT (2)*. Vol. 12106. LNCS. Springer, 2020, pp. 493–522. URL: https://ia.cr/2018/537.

[7] Wouter Castryck and Thomas Decru. "An Efficient Key Recovery Attack on SIDH". In: *EUROCRYPT (5)*. Vol. 14008. LNCS. Springer, 2023, pp. 423–447. URL: https://ia.cr/2022/975.

[8] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. "CSIDH: An Efficient Post-Quantum Commutative Group Action". In: *ASIACRYPT (3)*. Vol. 11274. LNCS. Springer, 2018, pp. 395–427. URL: https://ia.cr/2018/383.

[9] Kostas Kryptos Chalkias, Jonas Lindstrøm, and Arnab Roy. *An Efficient Hash Function for Imaginary Class Groups*. Preprint. 2024. URL: https://ia.cr/2024/295.

[10] Mingjie Chen, Antonin Leroux, and Lorenz Panny. "SCALLOP-HD: Group Action from 2-Dimensional Isogenies". In: *Public Key Cryptography (3)*. Vol. 14603. LNCS. Springer, 2024, pp. 190–216. URL: https://ia.cr/2023/1488.

[11] Leonardo Colò and David Kohel. "Orienting supersingular isogeny graphs". In: *NutMiC 2019*. 2019. URL: https://ia.cr/2020/985.

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

[13] Jean-Marc Couveignes. *Hard Homogeneous Spaces*. Preprint. 2006. URL: https://ia.cr/2006/291.

[14] Pierrick Dartois, Luciano Maino, Giacomo Pope, and Damien Robert. *An Algorithmic Approach to $(2,2)$-isogenies in the Theta Model and Applications to Isogeny-based Cryptography*. Preprint. URL: https://ia.cr/2023/1747.

[15] Luca De Feo, Tako Boris Fouotsa, Péter Kutas, Antonin Leroux, Simon-Philipp Merz, Lorenz Panny, and Benjamin Wesolowski. "SCALLOP: Scaling the CSI-FiSh". In: *Public Key Cryptography (1)*. Vol. 13940. LNCS. Springer, 2023, pp. 345–375. URL: https://ia.cr/2023/058.

[16] Luca De Feo, Tako Boris Fouotsa, and Lorenz Panny. "Isogeny Problems with Level Structure". In: *EUROCRYPT (6)*. Vol. 14656. LNCS. Springer, 2024, pp. 181–204. URL: https://ia.cr/2024/459.

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

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

[19] G. H. Hardy and E. M. Wright. *An Introduction to the Theory of Numbers*. 5th ed. Oxford University Press, 1979.

[20] David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, Vladimir Soukharev, and David Urbanik. *Supersingular Isogeny Key Encapsulation*. Submission to the NIST-PQC post-quantum standardization project. 2017. URL: https://sike.org.

[21] Michael J. Jacobson Jr. "Applying sieving to the computation of quadratic class groups". In: *Mathematics of Computation* 68.226 (1999), pp. 859–867.

[22] Ernst Kani. "The number of curves of genus two with elliptic differentials". In: *Journal für die reine und angewandte Mathematik (Crelles Journal)* 1997 (1997), pp. 122–93.

[23] Thorsten Kleinjung. "Quadratic sieving". In: *Mathematics of Computation* 85.300 (2016), pp. 1861–1873.

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

[25] Jonathan Komada Eriksen, Lorenz Panny, Jana Sotáková, and Mattia Veroni. "Deuring for the People: Supersingular Elliptic Curves with Prescribed Endomorphism Ring in General Characteristic". In: *LuCaNT 2023*. 2023. URL: https://ia.cr/2023/106.

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

[27] Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. "A Direct Key Recovery Attack on SIDH". In: *EUROCRYPT (5)*. Vol. 14008. LNCS. Springer, 2023, pp. 448–471. URL: https://ia.cr/2023/640.

[28] Hiroshi Onuki. "On oriented supersingular elliptic curves". In: *Finite Fields and Their Application* 69 (2021). URL: https://arxiv.org/abs/2002.09894.

[29] Aurel Page and Damien Robert. *Introducing Clapoti(s): Evaluating the isogeny class group action in polynomial time*. Preprint. 2023. URL: https://ia.cr/2023/1766.

[30] Lorenz Panny. "Cryptography on Isogeny Graphs". PhD thesis. Technische Universiteit Eindhoven, 2021. URL: https://yx7.cc/docs/phd/thesis.pdf.

[31] Lorenz Panny. *CSI-FiSh really isn't polynomial-time*. https://yx7.cc/blah/2023-04-14.html. Accessed: 2024-09-17. 2023.

[32] Chris Peikert. "He Gives C-Sieves on the CSIDH". In: *EUROCRYPT (2)*. Vol. 12106. LNCS. Springer, 2020, pp. 463–492.

[33] Christophe Petit and Spike Smith. "An improvement to the quaternion analogue of the ℓ-isogeny path problem". In: *MathCrypt 2018*. 2018. URL: https://crypto.iacr.org/2018/affevents/mathcrypt/medias/08-50_3.pdf.

[34] Srinivasa Ramanujan. "On the expression of a number in the form $ax^2 + by^2 + cz^2 + dw^2$". In: *Proceedings of the Cambrdige Philosophical Society* 19 (1917), pp. 11–21.

[35] Damien Robert. "Breaking SIDH in Polynomial Time". In: *EUROCRYPT (5)*. Vol. 14008. LNCS. Springer, 2023, pp. 472–503. URL: https://ia.cr/2022/1038.

[36] Alexander Rostovtsev and Anton Stolbunov. *Public-Key Cryptosystem Based on Isogenies*. Preprint. 2006. URL: http://ia.cr/2006/145.

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

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