# Qlapoti: Simple and Efficient Translation of Quaternion Ideals to Isogenies

Giacomo Borin[1,2], Maria Corte-Real Santos[3], Jonathan Komada Eriksen[4], Riccardo Invernizzi[4], Marzio Mula[5], Sina Schaeffler[1,6], and Frederik Vercauteren[4](✉)

[1] IBM Research Zurich, Zürich, Switzerland
ac25@gbor.in
[2] University of Zurich, Zürich, Switzerland
[3] ENS de Lyon, CNRS, UMPA, UMR 5669, Lyon, France
maria.corte_real_santos@ens-lyon.fr
[4] COSIC, KU Leuven, Leuven, Belgium
{jeriksen,riccardo.invernizzi,Frederik.Vercauteren}@esat.kuleuven.be
[5] University of the Bundeswehr Munich, Munich, Germany
marzio.mula@unibw.de
[6] ETH Zurich, Zürich, Switzerland
sschaeffle@ethz.ch

**Abstract.** The main building block in isogeny-based cryptography is an algorithmic version of the Deuring correspondence, called IdealToIsogeny. This algorithm takes as input left ideals of the endomorphism ring of a supersingular elliptic curve and computes the associated isogeny. Building on ideas from QFESTA, the Clapoti framework by Page and Robert reduces this problem to solving a certain norm equation. The current state of the art is however unable to efficiently solve this equation, and resorts to a relaxed version of it instead. This impacts not only the efficiency of the IdealToIsogeny procedure, but also its success probability. The latter issue has to be mitigated with complex and memory-heavy rerandomization procedures, but still leaves a gap between the security analysis and the actual implementation of cryptographic schemes employing IdealToIsogeny as a subroutine. For instance, in SQIsign the failure probability is still $2^{-60}$ which is not cryptographically negligible.

The main contribution of this paper is a very simple and efficient algorithm called Qlapoti which approaches the norm equation from Clapoti directly, solving all the aforementioned problems at once. First, it makes the IdealToIsogeny subroutine between 2.2 and 2.6 times faster. This significantly improves the speed of schemes using this subroutine, including notably SQIsign and PRISM. On top of that, Qlapoti has a cryptographically negligible failure probability. This eliminates the need for rerandomization, drastically reducing memory consumption, and allows for cleaner security reductions.

Author list in alphabetical order; see https://ams.org/profession/leaders/CultureStatement04.pdf.

G. Hanaoka and B.-Y. Yang (Eds.): ASIACRYPT 2025, LNCS 16248, pp. 174–205, 2026.
https://doi.org/10.1007/978-981-95-5113-2_6

**Keywords:** Isogeny-based cryptography · quaternion algebras · ideal to isogeny translation · SQIsign · PRISM

# 1 Introduction

Post-quantum cryptography aims to build cryptographic protocols that are secure against both classical and quantum adversaries. The growing investment in quantum computing has prompted NIST to seek replacements for classical public key cryptosystems, resulting in the recent NIST standards: Kyber [34] for key encapsulation and Dilithium [22], Falcon [31], and SPHINCS+ [17] as signature schemes. Despite this, NIST opened an alternate call for digital signatures schemes [27], highlighting the need for further research. Indeed, we currently rely mostly on lattice-based security assumptions, and the signature sizes are much larger than the pre-quantum signatures that are currently used.

Isogeny-based digital signatures are promising candidates, since they are very compact and are based on radically different hardness assumptions, providing needed diversity. The two most prominent isogeny-based signatures schemes are SQIsign [2] and PRISM [5], whose combined signature and public-key sizes are between 7.7 and 37 times smaller[1] than the already standardized protocols [30]. Due to these advantages, SQIsign has progressed to the second round in the call for additional signatures. However, the signing algorithm is significantly slower, and more complex compared to other schemes. Therefore, to make isogeny-based signatures practical, it is crucial to find new techniques that simplify these algorithms and improve their performance.

An essential part of many isogeny-based signature schemes is the *Deuring correspondence*, which gives a bijection between quaternion ideals and isogenies. After pioneering work by Kohel, Lauter, Petit, Tignol [20] introducing the KLPT algorithm, this correspondence was made effective (also due to Galbraith, Petit, Silva [16]), meaning that one could efficiently translate a quaternion ideal to its corresponding isogeny. This was subsequently used to construct the first version of SQIsign [13,14]. However, its reliance on precisely the (generalized) KLPT algorithm made the original signing procedure very slow and complex.

Inspired by the attacks on SIKE [8,23,32] (an earlier isogeny-based key-encapsulation mechanism), isogeny-based cryptography has been revolutionized by efficiently representing isogenies as components of higher-dimensional isogenies. For instance, using 2-dimensional isogenies, QFESTA [25] showed how to compute an isogeny of a given degree from a base curve $E_0$ with known endomorphism ring. This technique was then used by Page and Robert in Clapoti [28], who showed how to translate any (quadratic) ideal to its corresponding isogeny efficiently.

The performance of SQIsign was also majorly improved by leveraging higher-dimensional isogenies [6,11,26]. In particular, a technique based on Clapoti can be applied to quaternion ideals as well [6,26], which results in an algorithm to

[1] In more detail, for NIST Level I SQIsign is 7.7 times smaller than Falcon, 18 times smaller than ML-DSA and 37 times smaller than SPHINCS+.

efficiently translate ideals into the corresponding isogeny without reliance on the KLPT algorithm, by embedding it in a 2-dimensional isogeny. From now on, we refer to this algorithm as IdealToIsogeny as in [2, Alg. 3.13]. These same tools were also used to create the signature scheme PRISM, a simple hash-and-sign signature scheme with similar performance to SQIsign, but relying on a different hardness assumption.

Despite the huge improvements of IdealToIsogeny over previous KLPT-based algorithms, it is still the dominating cost in the key generation and signing procedures of SQIsign and PRISM. In particular, IdealToIsogeny takes up about 90% of the time in key generation and 80% in signing for SQIsign, at all different security levels.

**Overview of State-of-the-Art.** We now give an overview of the main ideas underlying the state-of-the-art in ideal-to-isogeny translation.

QFESTA: The current state-of-the-art builds on an insight due to QFESTA [25]: given an endomorphism $\theta \in \mathrm{End}(E_0)$ of degree $\deg(\theta) = d_1 \cdot d_2$ with $\gcd(d_1, d_2) = 1$, it is possible to *efficiently* split this endomorphism as long as $d_1 + d_2 = 2^e$ and $E_0[2^e]$ is defined over $\mathbb{F}_{p^2}$, in particular $2^e \leq p + 1$. Splitting an endomorphism means to write it as a composition of two isogenies: $\theta = \varphi_2 \circ \varphi_1 = \psi_1 \circ \psi_2$, where $\deg(\varphi_1) = \deg(\psi_1) = d_1$ and $\deg(\varphi_2) = \deg(\psi_2) = d_2$. Note that indeed we get two decompositions, depending on the order of the degrees $d_1$ and $d_2$. Kani [18, §2] showed how such decomposition corresponds to the diagram:

$$\begin{tikzcd}
E_0 \arrow[r, "\varphi_1"] \arrow[d, "\psi_2"] & E_1 \arrow[d, "\varphi_2"] \\
E_2 \arrow[r, "\psi_1"] & E_0
\end{tikzcd}$$

and that these isogenies define a 2-dimensional isogeny $\Phi : E_0 \times E_0 \to E_1 \times E_2$ of degree $d_1 + d_2 = 2^e$, whose kernel only depends on the action of $\theta$ on $E_0[2^e]$. Since $\theta$ is given, we can explicitly determine this kernel and efficiently compute the codomain $E_1 \times E_2$ using a chain of $(2, 2)$-isogenies. Note that this is already sufficient to generate an isogeny of a given degree $u$ say, as long as we can find an endomorphism $\theta$ of degree $\deg(\theta) = u(2^e - u)$. Finding such $\theta \in \mathrm{End}(E_0)$ is efficient [25, §3.1] as long as $u(2^e - u) > p$.

Clapoti: Although QFESTA gives a method to construct an isogeny of a certain degree, it does not allow to control the codomain curves $E_1, E_2$. In their Clapoti framework [28], Page and Robert showed how $\theta$ can be constructed, such that splitting $\theta$ solves the ideal-to-isogeny translation problem. In particular, given a left $\mathcal{O}_0$-ideal $J$, they show how to compute the corresponding isogeny $\varphi_J : E_0 \to E_1$ by constructing an appropriate $\theta$. Recall that an ideal $I$ equivalent to $J$ results in the same (up to isomorphism) codomain curve $E_1$, so if we take two ideals $I_1 = J\beta_1$ and $I_2 = J\beta_2$ both equivalent to $J$, the ideal $\overline{I}_2 \cdot I_1 = \overline{\beta}_2 \overline{J} J \beta_1$ is by construction principal, generated by an endomorphism $\theta$ of degree

$\mathrm{nrd}(I_1)\mathrm{nrd}(I_2)$. Thus as long as we can find equivalent ideals $I_1$ and $I_2$ whose norms satisfy

$$\mathrm{nrd}(I_1) + \mathrm{nrd}(I_2) = 2^e < p\,, \tag{1}$$

we can apply the QFESTA splitting technique to $\theta$ to compute $E_1$ (and also $\varphi_J$).

Unfortunately, the current state-of-the-art is unable to solve the above equation for $2^e < p$ directly. Instead, the current best approach is to generate random equivalent ideals $I_i$ of small norm and hope to find a solution to the more general equation

$$u \cdot \mathrm{nrd}(I_1) + v \cdot \mathrm{nrd}(I_2) = 2^e < p\,, \tag{2}$$

for integers $u, v$. Since the smallest norm of an equivalent ideal is in general around $\sqrt{p}$, and Eq. (2) typically only has solutions when $\mathrm{nrd}(I_1)\mathrm{nrd}(I_2) < 2^e < p$, the failure probability to find a solution to this equation is quite high, e.g., $2^{-8}$ in the basic version of SQIsign.

At a high level, there are currently two known approaches to lower the failure probability, as shown in PEGASIS [10]. The first is to take out the smooth-normed parts of $I_1$ and $I_2$, to lower $\mathrm{nrd}(I_1)$ and $\mathrm{nrd}(I_2)$. However, it is still unclear whether this approach can be applied to the quaternion setting, while staying in dimension 2. The second approach is a rerandomization procedure. Again, this becomes significantly more complex in the quaternion setting, and requires precomputing a list of different starting curves and connecting isogenies. This second approach is what is currently done in SQIsign, which lowers the failure probability to $\approx 2^{-60}$ [2, Table 8], but results in an algorithm that is not only significantly more complicated than the approach outlined above, but also uses a large amount of memory during execution [2, Section 3.2.5].

This leads to further issues. First, the fact that the failure probability is not negligible in the security parameter creates a bias in the distribution of signatures and public keys of the schemes relying on IdealToIsogeny. This bias cannot be ignored and is very hard to quantify for a rigorous security analysis, in particular for SQIsign, as noted in [3, §7]. Second, the high memory cost makes the current algorithms unsuitable for resource-constrained devices [1], despite the fact that small key and signature sizes make these platforms particularly appealing for deploying SQIsign and PRISM.

Finally, even when a solution $I_1, I_2$ is found, we still need to construct auxiliary isogenies of degrees $u$ and $v$, e.g., using the QFESTA approach. This means that, instead of a single chain of $(2,2)$-isogenies, three such chains must now be computed (with those corresponding to $u$ and $v$ being of half the length).

**Our Approach.** The main problem with the current approach is that it generates the ideals $I_1$ and $I_2$ independently and of small norm. This makes it virtually impossible to solve Equation (1) directly. Our approach constructs the ideals $I_1$ and $I_2$ simultaneously, of norm a bit smaller than $p$ (in contrast with the $\sqrt{p}$ of current methods), but such that the sum of their norms satisfies Eq. (1).

We start by replacing $J$ with an equivalent ideal $I$ of smallest norm $\mathrm{nrd}(I) = n$ and write it as $I = \mathcal{O}_0\langle n, \alpha\rangle$ with $\alpha$ a random generator of somewhat small norm. Since the ideals $I_1, I_2$ will also be equivalent to $I$, there exist $\beta_k \in I$ for $k = 1, 2$ such that $I_k = I\overline{\beta}_k/n$ [13, Lemma 1]. The norm equation for the ideals $I_i$ can thus be replaced by a norm equation for the elements $\beta_i$:

$$\operatorname{nrd}(\beta_1) + \operatorname{nrd}(\beta_2) = 2^e \cdot n\,. \tag{3}$$

Since $\beta_k \in I$, we can write them in full generality as $\beta_k = \gamma_k \cdot n + \gamma'_k \cdot \alpha$ for some $\gamma_k, \gamma'_k \in \mathcal{O}_0$. Note that each $\gamma$ has 4 unknown integral coefficients, so simply substituting these expressions in Eq. (3) would result in a quadratic form in 16 unknowns. Since this is a bit too complex, we simplify things by setting $\gamma'_k = 1$. Using the expression $\operatorname{nrd}(x + y) = \operatorname{nrd}(x) + \operatorname{tr}(x\bar{y}) + \operatorname{nrd}(y)$ and dividing by $n$, we can rewrite the above equation as:

$$n(\operatorname{nrd}(\gamma_1) + \operatorname{nrd}(\gamma_2)) + \operatorname{tr}((\gamma_1 + \gamma_2)\bar{\alpha}) = 2^e - 2r\,, \tag{4}$$

where $r$ is the integer such that $\operatorname{nrd}(\alpha) = n \cdot r$. Note that since $\alpha \in I$, we indeed have $n \mid \operatorname{nrd}(\alpha)$. As the right-hand side is smaller than $p$, we need to force the left-hand side also to be smaller than $p$. In particular, we should only consider $\gamma_k \in \mathbb{Z}[\mathbf{i}]$. Indeed, we have that $\operatorname{nrd}(\mathbf{j}) = \operatorname{nrd}(\mathbf{k}) = p$, which would already make $\gamma_k$ too large. Restricting to $\gamma_k = a_k + b_k\mathbf{i}$, and defining $\alpha = a_\alpha + b_\alpha\mathbf{i} + c_\alpha\mathbf{j} + d_\alpha\mathbf{k}$, we finally end up with the equation

$$n(a_1^2 + b_1^2 + a_2^2 + b_2^2) + 2a_\alpha(a_1 + a_2) + 2b_\alpha(b_1 + b_2) = 2^e - 2r\,. \tag{5}$$

To reduce the above equation to a standard sum-of-squares problem, we need to get rid of the terms coming from the trace. The crucial insight is that this can be accomplished by looking at the equation modulo $n$, finding a small solution $(s, t)$ to $2a_\alpha x + 2b_\alpha y = 2^e - 2r \bmod n$ and then imposing the linear conditions $a_1 + a_2 = s$ and $b_1 + b_2 = t$. Since these terms by definition are now constant, they can be moved to the right-hand side. By a final substitution of $a_2 = s - a_1$ and $b_2 = t - b_1$, dividing by $n$ and multiplying by 2, we recover a standard sum-of-squares problem

$$(2a_1 - s)^2 + (2b_1 - t)^2 = 2w/n - s^2 - t^2\,, \tag{6}$$

which can be solved using Cornacchia's algorithm [9]. Back substitution then gives $\beta_k$ and finally the ideals $I_1, I_2$.

**Our Contributions.** The main contribution of this paper is the algorithm sketched above, which we call Qlapoti. The efficient resolution of the norm equation Eq. (1) immediately implies a much faster IdealToIsogeny since we no longer require the auxiliary degree $u$ and $v$ isogenies, and therefore are left with computing only a single chain of $(2, 2)$-isogenies. A second contribution is the implementation of Qlapoti in SageMath and in C, which confirm a speed-up over IdealToIsogeny of up to a factor of 2.6. We further integrated our Qlapoti implementations into both SQIsign and PRISM, showing that we achieve significant improvements in both cases. These improvements are not only restricted to computational speed; apart from being faster, measurements show that running SQIsign with Qlapoti requires between a factor of 11 and 34 times less memory, compared to the NIST Round 2 submission [2]. Finally, our third contribution is a detailed statistical analysis of Qlapoti, resulting in accurate failure probabilities, under reasonable heuristics. For the three NIST levels, we achieve (heuristic) failure probabilities of $2^{-197}, 2^{-312}, 2^{-438}$, which are negligible compared to the respective security levels. This also makes the security proof of SQIsign and PRISM much cleaner since signature failures are no longer a problem.

The implementations can be found in our public repository:

https://github.com/KULeuven-COSIC/Qlapoti

## 2 Preliminaries

In this section we recall the effective Deuring correspondence, focusing on its application to isogeny-based signature schemes, namely higher-dimensional variants of SQIsign [2,6] and PRISM [5]. We assume the reader is familiar with the literature on elliptic curves and their isogenies. We refer the reader to [12,35] for more information. Throughout the paper, $p$ is a prime with $p \equiv 3 \pmod 4$, $E_0$ is the (supersingular) elliptic curve $y^2 = x^3 + x$ defined over $\mathbb{F}_p$, and $\mathcal{O}_0$ is the quaternion order isomorphic to $\mathrm{End}(E_0)$ (See Example 1).

### 2.1 Quaternion Algebras and Deuring Correspondence

Let $\mathcal{B}_{p,\infty}$ be the unique (up to isomorphism) *quaternion algebra* over $\mathbb{Q}$ ramified at $p$ and $\infty$. This means it is a division algebra defined by $\mathbb{Q}+\mathbb{Q}\mathbf{i}+\mathbb{Q}\mathbf{j}+\mathbb{Q}\mathbf{k}$, where $\mathbf{i}^2 = -1$, $\mathbf{j}^2 = -p$, $\mathbf{k} = \mathbf{ij} = -\mathbf{ji}$. For a given element $\alpha = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} \in \mathcal{B}_{p,\infty}$ we define its conjugate $\bar{\alpha} := a - b\mathbf{i} - c\mathbf{j} - d\mathbf{k}$ and its *reduced norm* $\mathrm{nrd}(\alpha) := \alpha\bar{\alpha}$. A *fractional ideal* in $\mathcal{B}_{p,\infty}$ is a $\mathbb{Z}$-submodule of rank 4. An *order* $\mathcal{O}$ is a fractional ideal that is also a subring. An order is *maximal* if it is not properly contained in any other order, and a maximal order $\mathcal{O}$ is *special extremal* if it contains a suborder $\mathbb{Z}[\omega] + \mathbb{Z}[\omega]\mathbf{j}$ such that $\omega$ has smallest norm in $\mathcal{O}$ and $\mathbb{Z}[\omega] \subset (\mathbb{Z}[\omega]\mathbf{j})^{\perp}$. Let $I$ be a fractional ideal in $\mathcal{B}_{p,\infty}$. We define the left order of $I$ to be $\mathcal{O}_L(I) := \{\alpha \in \mathcal{B}_{p,\infty} \mid \alpha I \subset I\}$. We can similarly define the right order $\mathcal{O}_R(I)$ of a fractional ideal $I$, and $I$ is called a *connecting ideal* for $\mathcal{O}_L(I)$ and $\mathcal{O}_R(I)$. If $I$ is contained in its left order (or, equivalently, in its right order) then it is an *integral ideal*, or just an *ideal* for short.

For a fractional ideal $I$, we denote its conjugate by $\bar{I} = \{\bar{\alpha} \mid \alpha \in I\}$. The reduced norm of an ideal $I$, denoted by $\mathrm{nrd}(I)$, is defined as the gcd of the reduced norms of the elements of $I$. For a maximal order $\mathcal{O}$, any left $\mathcal{O}$-ideal $I$ can be written as $I = \mathcal{O}\langle\alpha, \mathrm{nrd}(I)\rangle = \mathcal{O}\alpha + \mathcal{O}\mathrm{nrd}(I)$ for some $\alpha \in I$. Two ideals $I$ and $J$ are *equivalent* if there exists $\beta \in \mathcal{B}_{p,\infty}^{\times}$ such that $I = J\beta$. We denote equivalence by $I \sim J$. Also, by [13, Lemma 1], all the equivalent ideals $I \sim J$ are of the form $\chi_J(\alpha) := J\bar{\alpha}/\mathrm{nrd}(J)$ for some $\alpha \in J$ and $\alpha$ determines $I$ up to multiplication by an element of $\mathcal{O}_0^{\times}$. The norm of $I = \chi_J(\alpha) = J\bar{\alpha}/\mathrm{nrd}(J)$ is $\mathrm{nrd}(I) = \mathrm{nrd}(\alpha)/\mathrm{nrd}(J)$.

Since fractional ideals are also rank 4 $\mathbb{Z}$-lattices, elements of small norm can be efficiently found via classical lattice algorithms. In this way, we can define the procedure $\mathsf{SmallestEquivIdeal}$, that given as input an ideal of (large) norm $J$, finds the smallest element $\alpha \in J$ and returns the ideal $\chi_J(\alpha)$. The following lemma gives an upper bound for the norm of the smallest element in each equivalence class of ideals. This will prove useful when analysing the output size of our algorithm.

**Lemma 1 (Lemma 12 [11]).** *Let $\mathcal{O}$ be a maximal order and let $J$ be a left $\mathcal{O}$-ideal. Then there exists $\alpha \in J$ such that $\operatorname{nrd}(\alpha) \leq 2\sqrt{2p}/\pi \cdot \operatorname{nrd}(J)$ and thus $\operatorname{nrd}(\chi_J(\alpha)) \leq 2\sqrt{2p}/\pi$.*

Deuring [15] showed a categorical equivalence between maximal orders in $\mathcal{B}_{p,\infty}$ and supersingular elliptic curves defined over $\mathbb{F}_{p^2}$. This equivalence is known as the *Deuring correspondence*. Under this correspondence, to each maximal order $\mathcal{O}$ of $\mathcal{B}_{p,\infty}$ we can associate a supersingular elliptic curve $E$ over $\mathbb{F}_{p^2}$, up to $\bar{\mathbb{F}}_p$-isomorphism, such that $\operatorname{End}(E) \cong \mathcal{O}$. An isogeny $\varphi : E_1 \to E_2$ corresponds to an ideal $I_\varphi$, where $\mathcal{O}_L(I_\varphi) \cong \operatorname{End}(E_1)$ and $\mathcal{O}_R(I_\varphi) \cong \operatorname{End}(E_2)$. Moreover, $\deg(\varphi) = \operatorname{nrd}(I_\varphi)$.

A more detailed discussion of quaternion algebras and the Deuring correspondence can be found in [13,36].

*Example 1.* Since $p \equiv 3 \bmod 4$, the elliptic curve $E_0 : y^2 = x^3 + x$ defined over $\mathbb{F}_{p^2}$ is supersingular. We can define endomorphisms $\iota : (x,y) \mapsto (-x, \sqrt{-1}y)$ and $\pi : (x,y) \mapsto (x^p, y^p)$ of $E_0$, where $\sqrt{-1}$ is a fixed square root of $-1$ in $\mathbb{F}_{p^2}$. We have the following isomorphism of rings:

$$\begin{aligned} \mathcal{O}_0 := \mathbb{Z}\left\langle 1, \mathbf{i}, \frac{\mathbf{i}+\mathbf{j}}{2}, \frac{1+\mathbf{k}}{2} \right\rangle &\longrightarrow \operatorname{End}(E_0), \\ a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} &\longmapsto a + b\iota + c\pi + d\iota\pi. \end{aligned}$$

Notably, $\mathcal{O}_0$ is a special extremal order since it is maximal and it contains $\mathbb{Z}[\mathbf{i}] + \mathbb{Z}[\mathbf{i}]\mathbf{j}$.

### 2.2 Kani's Lemma

Kani's lemma [18] gives a criterion to compute isogenies of dimension one using isogenies of dimension two. It was at the heart of the SIDH attacks [8,23,32], but it quickly became an indispensable constructive tool for isogeny-based protocols. Our formulation of Kani's lemma follows [23].

**Theorem 1 (Kani).** *Let $d_1, d_2$ and $N$ be pairwise coprime integers such that $N = d_1 + d_2$, and let $E_0$, $E_1$, $E_2$, and $E_3$ be elliptic curves connected by the following commutative diagram of isogenies:*

$$\begin{tikzcd} E_0 \arrow[r, "\varphi_1"] \arrow[d, "\psi_2"] & E_1 \arrow[d, "\varphi_2"] \\ E_2 \arrow[r, "\psi_1"] & E_3 \end{tikzcd}$$

*such that* $\deg(\varphi_1) = \deg(\psi_1) = d_1$, $\deg(\varphi_2) = \deg(\psi_2) = d_2$ *and* $\varphi_2 \circ \varphi_1 = \psi_1 \circ \psi_2$*. Then the map*

$$\Phi = \begin{pmatrix} \varphi_1 & \widehat{\varphi_2} \\ -\psi_2 & \widehat{\psi_1} \end{pmatrix} : E_0 \times E_3 \to E_1 \times E_2$$

*is an isogeny of (principally polarized) abelian varieties with kernel*

$$\ker(\Phi) = \{([d_1]P, \varphi_2 \circ \varphi_1(P)) \mid P \in E_0[N]\} \cong \frac{\mathbb{Z}}{N\mathbb{Z}} \times \frac{\mathbb{Z}}{N\mathbb{Z}}.$$

Assuming that $N$ is smooth and all $N$-torsion points are rational (in our case, we have $N = 2^e$), the isogeny $\Phi$ can be efficiently evaluated at any point on $E_0 \times E_3$. Indeed, to evaluate the isogeny $\varphi_1$ at any point $P \in E_0$ we evaluate $\Phi$ on $(P, \infty_{E_3})$ and then project the result onto $E_1$. In this way, the generators of the kernel defining $\Phi$ encode an *efficient two-dimensional representation* of $\varphi_1$, as defined in [33].

### 2.3 Ideal to Isogeny Translation

A central task for isogeny-based schemes is the following: given a maximal order $\mathcal{O} \subset \mathcal{B}_{p,\infty}$ isomorphic to $\mathrm{End}(E)$ and a left $\mathcal{O}$-ideal $J$, translate $J$ to the isogeny associated to it via the Deuring correspondence. For ease of presentation, we limit this exposition to the case where the domain curve has $j$-invariant 1728; see Example 1. This core problem is the main bottleneck in key generation and signing procedure of the signature schemes SQIsign [2] and PRISM [5].

In [20], Kohel, Lauter, Petit and Tignol introduced the KLPT algorithm, which solves a "constrained norm equation". Namely, they develop an algorithm to find an $\alpha \in J$ such that $\mathrm{nrd}(\alpha) = \mathrm{nrd}(J)2^r$, where $r$ is a large positive integer. The equivalent ideal $\chi_J(\alpha)$ then has norm $2^r$, and thus, using auxiliary odd-degree isogenies, can be translated into a chain of 2-isogenies in polynomial time [13,14]. This algorithm is the core of the original SQIsign protocol [13]. However, in practice this translation strategy is inefficient due to the size of $r$ and the use of odd-degree isogenies.

We highlight the following two subroutines of the KLPT algorithm. The first one will be needed in the algorithm introduced in Sect. 3, while the second will only be used in references to earlier work:

- Cornacchia: given two integers $d, n$, outputs a solution $(x, y)$ to $x^2 + dy^2 = n$, when such a solution exists. Otherwise, it outputs $\perp$. A necessary condition for the existence of a solution is that $-d$ is a quadratic residue modulo $n$. The algorithm can be traced back to Cornacchia [9,24] and pseudocode can be found in the SQIsign specifications [2, Algorithm 3.11].
- RepresentInteger: given $M \in \mathbb{N}$ where $M > p$, outputs an element $\gamma \in \mathcal{O}_0$ of norm $M$, where $\mathcal{O}_0$ is the order defined in Example 1. It relies on Cornacchia's algorithm and can be generalized to all special extremal orders. The first version of this algorithm was introduced at the same time as KLPT [20], and a variant that works for generic orders was subsequently proposed in [14].

**Translations Using Higher Dimensional Isogenies.** Page and Robert [28] recently introduced Clapoti, a procedure that leverages higher dimensional isogenies and Kani's lemma (see Theorem 1) to obtain a new algorithmic tool for converting any ideal to the corresponding isogeny. While the Clapoti procedure in [28] is originally designed for ideals of quadratic imaginary orders, its core ideas can be adapted to efficiently translate (almost) any left $\mathcal{O}_0$-ideal, when working over $\mathbb{F}_{p^2}$ with $p = f2^{e+2} - 1$, for some small odd $f > 0$. This is shown in the IdealToIsogeny algorithm from SQIsign2D-West [6], which we summarise in Algorithm 1 (Fig. 1).

```
Algorithm 1. IdealToIsogeny($J$)
Input: Left $\mathcal{O}_0$-ideal $J$, a basis $P_0, Q_0 \in E_0[2^e]$
Output: Isogeny $\varphi_J$ associated to $J$ and its domain $E_J$.
 1: Find $I_1, I_2 \sim J$ of coprime norms $d_1, d_2$ and $u, v$ such that $ud_1 + vd_2 = 2^e$, with
    $ud_1$ coprime to $vd_2$.                                              ▷ see [6, Algorithm 2]
 2: Get $\theta := \widehat{\varphi_{I_2}} \circ \varphi_{I_1}$;          ▷ Given by the principal ideal $I = \overline{I_2} \cdot I_1$
 3: Compute $\varphi_u, \varphi_v$ of degrees $u, v$ on $E_0[2^e]$;
 4: Get $P_u, Q_u \leftarrow \varphi_u(P_0), \varphi_u(Q_0)$;
 5: Set $\varphi \leftarrow \varphi_v \circ \theta \circ \widehat{\varphi_u}$;
 6: Set $K_P \leftarrow ([d_1 u]P_u, \varphi(P_u))$;
 7: Set $K_Q \leftarrow ([d_1 u]Q_u, \varphi(Q_u))$;
 8: Compute $\Phi : E_u \times E_v \to E_J \times E'$ of kernel $\langle K_P, K_Q \rangle$;   ▷ using Theorem 1
 9: Evaluate $\varphi_{I_1}$ on $E_0[2^e]$ using $\Phi(P_u, \infty_{E_v}) = ([u]\varphi_{I_1}(P_0), *)$ and $\Phi(Q_u, \infty_{E_v}) =$
    $([u]\varphi_{I_1}(Q_0), *)$;
10: Using $\varphi_{I_1}$, evaluate $\varphi_J = [d_1^{-1}]\varphi_{I_1} \circ \beta_1$ on $E_0[2^e]$;   ▷ using [6, Lemma 11]
11: return $\varphi_J, E_J$;
```

$$\begin{tikzcd}
E_u \arrow[r, "\widehat{\varphi_u}"] \arrow[dd, "\psi_{v,2}"] \arrow[ddrr, "\varphi"] & E_0 \arrow[r, "\varphi_{I_1}"] \arrow[dr, "\theta"] & E_J \arrow[d, "\widehat{\varphi_{I_2}}"] \\
 & & E_0 \arrow[d, "\varphi_v"] \\
E' \arrow[rr, "\psi_{u,1}"] & & E_v
\end{tikzcd}$$

**Fig. 1.** Diagram from [6], the isogenies $\psi_{u,1}$ and $\psi_{v,2}$ are the isogenies given by the relative pushforwards of the corresponding isogenies.

Step 1 consists of sampling elements $\beta_1$ and $\beta_2$ in $J$ until the (reduced) norms $d_1, d_2$ of $I_1 = \chi_J(\beta_1)$, $I_2 = \chi_J(\beta_2)$ satisfy the equation $ud_1 + vd_2 = 2^e$ for some $u, v > 0$. Note that since $I_1, I_2$ are left $\mathcal{O}_0$-ideals equivalent to $J$, the composition

$\widehat{\varphi_{I_2}} \circ \varphi_{I_1}$ is an endomorphism $\theta$ of $E_0$, which can be computed from $\overline{\beta_2}\beta_1$ via the isomorphism in Example 1 (see Step 2).

For Step 3, different strategies can be used depending on $u, v$. In [6], $u, v \approx \sqrt{p}$ and so $\varphi_u$ and $\varphi_v$ can be sampled and computed via a direct application of the QFESTA algorithm [25]. This requires that the endormorphism ring of $E_0$ is isomorphic to a special extremal order. In some special cases – e.g., if $u, v$ are squares, sum of squares or smooth integers – we can sample $\varphi_u$ and $\varphi_v$ using more efficient representations.

Step 8, inspired by [28], consists of embedding the isogenies $\varphi_1 \circ \widehat{\varphi}_u$ and $\varphi_2 \circ \widehat{\varphi}_v$, of degrees $d_1 u$ and $d_2 v$ respectively, into a $(2^e, 2^e)$-isogeny using Kani's lemma. Indeed, by applying Theorem 1, we obtain an efficient representation of the isogeny $\varphi_1 \circ \widehat{\varphi}_u$. By composing with $\varphi_u$, we can use this to evaluate points for $\varphi_{I_1}$, and thus recover $\varphi_J$ by applying [6, Lemma 11] (see Step 10).

A more detailed description of each step of the algorithm can be found in [6, §4.2] and [2].

**Limitations.** As Algorithm 1 is the main algorithmic building block used in key generation and signing procedure in SQIsign, extensive analysis has been performed to understand its failure probability [2, §9.3]. There are three possible reasons why Algorithm 1 may fail. Two of them are well understood: failures in the computations of the higher-dimensional isogenies and failures in the subroutine RepresentInteger, which is necessary to compute the auxiliary isogenies $\varphi_u, \varphi_v$ in Step 3.

However, it is important to note that the failure probability of Step 1 is not well understood [2, Section 9.3.2]. Let $d_1 = \mathrm{nrd}(\beta_1)/\mathrm{nrd}(J)$ and $d_2 = \mathrm{nrd}(\beta_2)/\mathrm{nrd}(J)$. The main obstacle in finding suitable $\beta_1, \beta_2$ is that the existence of $u, v$ satisfying $ud_1 + vd_2 = 2^e$ is only guaranteed when $(d_1 - 1)(d_2 - 1) < 2^e + 1$. Unfortunately, this condition is rarely met in our setting, since $d_1, d_2 \approx \sqrt{p}$ and $2^e = (p-1)/f$. If this fails, the probability of finding a valid pair decreases rapidly – approximately as $2^e/(d_1 d_2)$. For this reason, $\beta_1$ and $\beta_2$ have to be selected among the shortest vectors of $J$, whose expected norms are approximately $\sqrt{p}$. However, for the lattices we consider we cannot expect the norms of the shortest vectors to behave as random numbers – for instance, the first two successive minima in $\mathcal{O}_0$ are always equal because $\mathbf{i} \in \mathcal{O}_0$. As reported in [2, Section 9.3.2], for NIST Level I, the failure probability measured from extensive experiments is $2^{-8.2}$.

In [2, Section 3.2.3], a modified version of Algorithm 1, and in particular of Step 1, is presented to partially overcome this limitation, trading efficiency for a lower failure probability. The core idea is to provide additional randomness to this step by considering other possible starting curves than $E_0$ (precisely 6), having special extremal orders as endomorphism rings. Then they consider short elements not only in $J$, but also in the connecting ideals between the other orders and $\mathcal{O}_R(J)$. This, under another heuristic assumption regarding the distribution of these ideals, gives an estimated failure probability below $2^{-60}$. However, this not only slows down the algorithm, but also increases the memory requirements for the sieving procedure. Also, to avoid costly computations at runtime, these curves and data associated to them need to be precomputed. The successive steps of the function need to be adapted to handle the additional curves, increasing the overall complexity of the algorithm. Furthermore, even if small, the failure probability is non-negligible. This poses a significant limitation to the security analysis of the (signature) schemes that rely on the $\mathsf{IdealToIsogeny}$ procedure, since the failures introduce biases in the output distributions – a factor that is difficult to account for in the security analysis. In the case of $\mathsf{IdealToIsogeny}$, this issue is even more problematic, as the failure probability itself is difficult to estimate. Notably, for $\mathsf{SQIsign}$, this problem has already been observed in the literature [3, §7].

### 2.4 Isogeny-Based Signatures

We now detail the two main isogeny-based signature schemes impacted by the results of this paper: $\mathsf{SQIsign}$ and $\mathsf{PRISM}$. Though they follow two different frameworks (namely, one uses the Fiat-Shamir paradigm, the other is a hash-and-sign scheme), they both share the same underlying subroutines. In particular, they both strongly rely on the $\mathsf{IdealToIsogeny}$ algorithm described in Sect. 2.3. For the two schemes $p = f2^{e+2} - 1$ for some small odd $f > 0$ and $e$ a positive integer. The public key consists of a supersingular curve $E_{\mathsf{vk}}$, which can be represented by its $j$-invariant (in $2\log(p)$ bits) and two torsion points $P_{\mathsf{vk}}, Q_{\mathsf{vk}}$ generating $E_{\mathsf{vk}}[2^{e+2}]$. The latter can be deterministically generated, so the final size of the public key is $2\log(p)$ bits. The secret key consists of a left $\mathcal{O}_0$-ideal $I_{\mathsf{sk}}$ that is converted to an isogeny $\phi_{\mathsf{sk}} : E_0 \to E_{\mathsf{vk}}$ using the $\mathsf{IdealToIsogeny}$ algorithm. We briefly detail here the signing procedures of the two schemes, focusing on the role of ideal translations.

**SQIsign.** The $\mathsf{SQIsign}$ signature scheme is based on the following $\Sigma$-protocol[2].

- **Commitment**: the prover samples a random left $\mathcal{O}_0$-ideal $I_{\mathsf{com}}$ and uses $\mathsf{IdealToIsogeny}$ to compute the associated isogeny $\psi_{\mathsf{com}} : E_0 \to E_{\mathsf{com}}$. The commitment $E_{\mathsf{com}}$ is sent to the verifier.
- **Challenge**: the verifier samples a random power-of-2 degree isogeny $\phi_{\mathsf{chall}} : E_{\mathsf{vk}} \to E_{\mathsf{chall}}$ and sends it to the prover.
- **Response**: the prover translates $\phi_{\mathsf{chall}}$ to a left $\mathcal{O}_R(I_{\mathsf{sk}})$-ideal $I_{\mathrm{chall}}$ and uses it to find the ideal $J'$ associated to the isogeny $\phi_{\mathsf{chall}} \circ \phi_{\mathsf{sk}} \circ \widehat{\psi_{\mathsf{com}}} : E_{\mathsf{com}} \to E_{\mathsf{chall}}$. Using Lemma 1, the prover finds an equivalent ideal $I_{\mathsf{resp}}$ of small norm. Then, after factoring out the even-degree part, the prover computes the isogeny $\sigma_{\mathsf{resp}} : E_{\mathsf{com}} \to E_{\mathsf{chall}}$. The prover sends an efficient representation of the isogeny $\sigma_{\mathsf{resp}}$ to the verifier. To get an efficient representation that solely

[2] There exist several variants of the protocol, sharing the same intuitions; we present the one introduced by Basso, Dartois, De Feo, Leroux, Maino, Pope, Robert, and Wesolowski [6], which is also the one implemented in the submission to the NIST standardization [2].

relies on 2-dimensional isogenies, an additional auxiliary isogeny needs to be provided. This isogeny is obtained by translating a uniformly sampled ideal of suitable norm, again via the IdealToIsogeny algorithm.
- **Verification**: the verifier checks that $\sigma_{\mathsf{resp}}$ is a valid isogeny from $E_{\mathsf{com}}$ to $E_{\mathsf{chall}}$ of the expected degree.

The SQIsign signature scheme is obtained by applying the Fiat-Shamir paradigm to the previous $\Sigma$-protocol. Thus, every signature generation requires two executions of the IdealToIsogeny algorithm.

A complete proof of security of SQIsign is given in a recent work by Aardal, Basso, De Feo, Patranabis, and Wesolowski [3]. The protocol is sound given the hardness of the One Endomorphism Ring Problem for the supersingular elliptic curve $E_{\mathsf{vk}}$ [29]. The honest-verifier zero-knowledge property is instead more challenging to prove due to the potentially large degree of the response isogeny $\sigma_{\mathsf{resp}}$. The issue is that simulating valid transcripts requires computing high-dimensional representations without knowing the secret key (i.e. the endomorphism ring of $E_{\mathsf{com}}$) – a task that is currently considered computationally hard. The two possible workarounds to prove zero knowledge involve either *ad hoc* oracles – as done, e.g., in [6] – or variants of the Endomorphism Ring Problem and Fiat-Shamir heuristic in which additional hints are provided [3].

There is however still a gap between the security analysis they provide and the actual security of the SQIsign implementation, due to the non-negligible failure probability of the IdealToIsogeny algorithm. As stated in [3], developing a new IdealToIsogeny algorithm which has negligible failure probability closes this gap [3, Section 7].

**PRISM.** The PRISM signature scheme [5] is an isogeny-based hash-and-sign signature scheme that relies on the hardness of computing large-degree isogenies originating from elliptic curves whose endomorphism ring is unknown. As in SQIsign, PRISM uses higher-dimensional representations of isogenies both for the generation and the verification of signatures. Let $a > 0$ be a fixed integer smaller than $e$. Given an input message mes, the signing algorithm works as follows:

1. The message mes is hashed to a prime $q$ of exactly $a$ bits.
2. A left $\mathcal{O}_R(I_{\mathsf{sk}})$-ideal $I_\sigma$ of norm $q(2^a - q)$ is sampled uniformly.
3. Using IdealToIsogeny, the isogeny associated to $I_\sigma \cdot I_{\mathsf{sk}}$ is computed, and used to extract the corresponding isogeny $\sigma : E_{\mathsf{vk}} \to E_{\mathsf{sig}}$ of degree $q(2^a - q)$.
4. An efficient representation of the isogeny $\sigma$ is returned as a signature for mes.

The verification algorithm consists of checking that the signature is a valid isogeny of degree $q(2^a - q)$ from $E_{\mathsf{vk}}$. In [5, §4] the authors show that, in the random oracle model (ROM), the unforgeability of the scheme reduces to the hardness of the following:

*Problem 1.* Given a random curve $E$, a set of $N$ isogenies $\{\phi_i : E \to E_i\}_{i=1}^{N}$ of degree $q_i(2^a - q_i)$ for $q_i$ uniformly random in the set of primes of $a$ bits and $\phi_i$ uniformly random among the isogenies of degree $q_i(2^a - q_i)$, and a prime $\bar{q}$ of $a$ bits not in $\{q_i\}_{i=1}^{N}$, give an efficient representation of an isogeny of degree $\bar{q}(2^a - \bar{q})$.

We note that this reduction implicitly assumes that the signatures are distributed uniformly random among the isogenies of the same degree. However, in practice, this is not the case, since the IdealToIsogeny algorithm (used once during signing) creates a non-negligible bias in the distribution of the signatures, due to the failure probability.

## 3 Qlapoti: the Main Algorithm

In this section, we describe Qlapoti in detail following the outline sketched in the introduction. In particular, given as input a left $\mathcal{O}_0$-ideal $J$ it computes two equivalent ideals $I_1, I_2$ such that

$$\mathrm{nrd}(I_1) + \mathrm{nrd}(I_2) = 2^e \tag{7}$$

with $2^e < p$.

At the start of Qlapoti, we use the function SmallestEquivIdeal to replace $J$ by an equivalent ideal $I$ of minimal norm $n = \mathrm{nrd}(I)$ which by Lemma 1 satisfies $n \leq 2\sqrt{2p}/\pi$ and write $I = \mathcal{O}_0\langle n, \alpha\rangle$ with $\alpha$ a small random generator. Note that $\alpha$ is far from unique; in fact, there are exponentially many (in $\log n$) choices for $\alpha$ that will work (see the analysis in Sect. 4), which allows us to randomize the algorithm keeping the ideal $I$, and thus its norm $n$, fixed. Since $n \mid \mathrm{nrd}(\alpha)$, we define $r$ to be an integer satisfying $\mathrm{nrd}(\alpha) = n \cdot r$.

Since all ideals equivalent to $I$ are of the form $\chi_I(\beta) = I\bar{\beta}/\mathrm{nrd}(I)$ with $\beta \in I$ [13], it is thus sufficient to find two elements $\beta_1, \beta_2 \in I$ with

$$\mathrm{nrd}(\beta_1) + \mathrm{nrd}(\beta_2) = 2^e \cdot n\,. \tag{8}$$

We can write $\beta_k$ for $k = 1, 2$ as $\beta_k = \gamma_k \cdot n + \gamma'_k \cdot \alpha$ for some $\gamma_k, \gamma'_k \in \mathcal{O}_0$. However, we will only consider $\beta_k$ for which $\gamma'_k = 1$ and $\gamma_k \in \mathbb{Z}[\mathbf{i}]$. Although this choice seems to considerably limit the degrees of freedom in finding good $\beta_k$, it drastically simplifies (8), without compromising our ability to solve it.

To solve (8) we write $\alpha = a_\alpha + b_\alpha\mathbf{i} + c_\alpha\mathbf{j} + d_\alpha\mathbf{k}$ with $a_\alpha, b_\alpha, c_\alpha, d_\alpha \in \frac{1}{2}\mathbb{Z}$ and $\gamma_k = a_k + b_k\mathbf{i}$ for $k = 1, 2$ with $a_k, b_k \in \mathbb{Z}$. Note that both $2a_\alpha, 2b_\alpha \in \mathbb{Z}$ and we can assume these are reduced modulo $n$ and thus smaller than $n$, since $n \in I$. Then expanding (8) and dividing by $n$ gives

$$n(a_1^2 + b_1^2 + a_2^2 + b_2^2) + 2a_\alpha(a_1 + a_2) + 2b_\alpha(b_1 + b_2) = 2^e - 2r\,. \tag{9}$$

Reducing the above equation mod $n$ then gives

$$2a_\alpha(a_1 + a_2) + 2b_\alpha(b_1 + b_2) = 2^e - 2r \bmod n\,. \tag{10}$$

If we assume that either $\gcd(2a_\alpha, n) = 1$ or that $\gcd(2b_\alpha, n) = 1$, the above equation will have a solution (in fact it suffices that $\gcd(a_\alpha, b_\alpha, n) \mid 2^e - 2r$). By relabeling if necessary, we assume that $\gcd(2a_\alpha, n) = 1$. If we then also assume that $n \nmid 2b_\alpha$, the following lemma shows we expect to find at least a solution $(s, t)$ to the equation $2a_\alpha x + 2b_\alpha y = 2^e - 2r \bmod n$ whose 2-norm is in $O(\sqrt{n})$.

**Lemma 2.** *Let $a, b, c, n \in \mathbb{Z}$ with $\gcd(a, n) = 1$, $n \nmid b$ and $n > 0$, then over the choices of such $a, b, c \in \mathbb{Z}/n\mathbb{Z}$, the expected 2-norm of the smallest solution $(s, t)$ to $ax + by = c \bmod n$ can be bounded by $1.3278 \cdot \sqrt{n}$.*

*Proof.* Since $\gcd(a, n) = 1$, the integer solutions of the homogeneous equation $ax+by = 0 \bmod n$ form a rank-2 lattice $\Lambda$ of volume $n$, which follows easily from the basis vectors $(-ba^{-1} \bmod n, 1)$ and $(n, 0)$. Finding a small solution $(s, t)$ to the equation $ax + by = c \bmod n$ is now equivalent to solving a CVP problem for $\Lambda$ with target vector $(c', 0)$ where $c' = a^{-1} \cdot c \bmod n$. Let $B = \{\boldsymbol{b_1}, \boldsymbol{b_2}\}$ be a shortest basis of $\Lambda$ with lengths the successive minima $\lambda_1$ and $\lambda_2$. Then, an approximate solution to CVP can be found by Babai's rounding algorithm, i.e. write $(c', 0) = \alpha\boldsymbol{b_1} + \beta\boldsymbol{b_2}$ with $\alpha, \beta \in \mathbb{R}$, and round $\alpha$ and $\beta$ to the nearest integers. This solution satisfies

$$\|(c', 0) - (\lfloor\alpha\rceil\boldsymbol{b_1} + \lfloor\beta\rceil\boldsymbol{b_2})\|_2 = \|(\alpha - \lfloor\alpha\rceil)\boldsymbol{b_1} + (\beta - \lfloor\beta\rceil)\boldsymbol{b_2}\|_2 \leq \frac{\lambda_1 + \lambda_2}{2}.$$

To conclude, we use that the expected lengths of the successive minima of a random 2-dimensional lattice of volume $n$ can be estimated as $\lambda_1 \sim 0.6826\sqrt{n}$ and $\lambda_2 \sim 1.97314\sqrt{n}$ as shown in [4, Thm 13]. □

The above lemma gives a rather crude upper bound on the expected length of the shortest $(s, t)$. In Sect. 4, we provide a more in-depth analysis using the probability mass function of the random variable $\epsilon := \sqrt{s^2 + t^2}/\sqrt{n}$.

Once we have found a shortest solution $(s, t)$, we impose the linear relations $a_1 + a_2 = s$ and $b_1 + b_2 = t$ between the variables $a_k$ and $b_k$. By doing so, we make the sums constant, and the term $2a_\alpha s+2b_\alpha t$ can therefore be moved to the right hand side of Eq. (9). By construction of $(s, t)$, $w := 2^e - 2r - 2a_\alpha s - 2b_\alpha t$ is divisible by $n$ and Eq. (8) becomes $a_1^2 + a_2^2 + b_1^2 + b_2^2 = w/n$. Substituting $a_2 = s - a_1$ and $b_2 = t - b_1$ gives $2a_1^2 - 2a_1 s + s^2 + 2b_1^2 - 2b_1 t + t^2 = w/n$, which, upon multiplication by 2, finally results in

$$(2a_1 - s)^2 + (2b_1 - t)^2 = 2w/n - s^2 - t^2\,. \tag{11}$$

If $z := 2w/n - s^2 - t^2 > 0$ and if $z$ can be written as a sum of squares $z = z_0^2 + z_1^2$ (in particular, all prime factors $p_i$ of $z$ that are $p_i = 3 \bmod 4$ should occur with an even exponent), we can compute $z_0$ and $z_1$ using Cornacchia's algorithm [9]. Given $z_0$ and $z_1$, we then have to solve for the integers $a_1, b_1$ such that $2a_1 - s = z_0$ and $2b_1 - t = z_1$, which is only possible when $z_0 = s \bmod 2$ and $z_1 = t \bmod 2$. Note that it is possible to verify whether such solutions exist before we call Cornacchia, by looking at Eq. (11) modulo 4 and conclude that:

- If $z = 0 \bmod 4$, then $s = t = 0 \bmod 2$ is required.
- If $z = 1 \bmod 4$, then $s \neq t \bmod 2$ is required.
- If $z = 2 \bmod 4$, then $s = t = 1 \bmod 2$ is required.
- If $z = 3 \bmod 4$, no integral solutions exist.

If either $z$ cannot be written as a sum of squares, or $z, s, t$ do not satisfy the above equations, we simply sample a new small generator $\alpha$ and start over.

Given a compatible solution $(z_0, z_1)$, we compute $a_1 = (z_0 + s)/2$ and $b_1 = (z_1 + t)/2$ and also, $a_2 = s - a_1$ and $b_2 = t - b_1$. Finally, this gives $\beta_1 = n(a_1 + b_1\mathbf{i}) + \alpha$, $\beta_2 = n(a_2 + b_2\mathbf{i}) + \alpha$ and the equivalent ideals $I_1 = \chi_I(\beta_1)$ and $I_2 = \chi(\beta_2)$. The resulting algorithm is summarized in Algorithm 2.

**Algorithm 2.** $\mathsf{Qlapoti}(J)$

**Input:** Left $\mathcal{O}_0$-ideal $J$.
**Output:** Equivalent ideals $I_1$ and $I_2$ with $\mathrm{nrd}(I_1) + \mathrm{nrd}(I_2) = 2^e$.

```
 1: $I \leftarrow \mathsf{SmallestEquivIdeal}(J)$;                 ▷ Lemma 11,12 [11]
 2: $n \leftarrow \mathrm{nrd}(I)$;
 3: $\alpha \leftarrow \mathsf{SmallGenerator}(I)$;   ▷ write $\alpha = a_\alpha + b_\alpha\mathbf{i} + c_\alpha\mathbf{j} + d_\alpha\mathbf{k}$ for $a_\alpha, b_\alpha, c_\alpha, d_\alpha \in \frac{1}{2}\mathbb{Z}$
 4: $r \leftarrow \mathrm{nrd}(\alpha)/n$;
 5: if $\gcd(2a_\alpha, n) \neq 1$ and $\gcd(2b_\alpha, n) \neq 1$ then goto line 3
 6: Compute shortest $(s, t)$ such that $2a_\alpha s + 2b_\alpha t = 2^e - 2r \mod n$   ▷ See Lemma 2

 7: Set $z := 2(2^e - 2r - 2a_\alpha s - 2b_\alpha t)/n - s^2 - t^2$;
 8: if $z < 0$ then goto line 3
 9: if $z \equiv 0 \pmod 4$ and not $s = t = 0 \mod 2$ then goto line 3
10: if $z \equiv 1 \pmod 4$ and not $s \neq t \bmod 2$ then goto line 3
11: if $z \equiv 2 \pmod 4$ and not $s = t = 1 \mod 2$ then goto line 3
12: if $z \equiv 3 \pmod 4$ then goto line 3
13: $\mathsf{sol} \leftarrow \mathsf{Cornacchia}(z)$                ▷ May return $\perp$ if no solution is found
14: if $\mathsf{sol} = \perp$ then goto line 3
15: $z_0, z_1 \leftarrow \mathsf{sol}$                              ▷ $z_0^2 + z_1^2 = z$
16: if $z_0 \not\equiv s \pmod 2$ then swap $z_0, z_1$
17: $a_1 \leftarrow (z_0 + s)/2$, $b_1 \leftarrow (z_1 + t)/2$;
18: $a_2 \leftarrow s - a_1$, $b_2 \leftarrow t - b_1$
19: $\beta_1 = n(a_1 + b_1\mathbf{i}) + \alpha$, $\beta_2 = n(a_2 + b_2\mathbf{i}) + \alpha$;
20: $I_1 = \chi_I(\beta_1)$, $I_2 = \chi_I(\beta_2)$
21: return $I_1$, $I_2$;
```

## 4 Statistical Analysis and Failure Probability of Qlapoti

In this section, we provide an in-depth statistical analysis of the failure probability of Qlapoti (Algorithm 2). To this end we compute the expected number of $\alpha$'s required for Qlapoti to terminate. We refer to Sect. 5 for the statistics obtained by our implementation.

We first study the following two random variables:

- let $\delta := n/\sqrt{p}$ be the random variable over the domain of all ideal classes $[I]$ with $I \subset \mathcal{O}_0$, where $n$ denotes the norm of the smallest equivalent ideal in the class $[I]$,
- for fixed $n$, let $\epsilon_n := \sqrt{s^2+t^2}/\sqrt{n}$ denote the random variable over the domain $a, b, c \in \mathbb{Z}/n\mathbb{Z}$ with $\gcd(a, n) = 1$ and $n \nmid b$, and where $(s, t)$ is a solution of smallest norm to the linear equation $ax + by = c \bmod n$.

For a discrete random variable $X$, we denote with $f_X$ the probability mass function and with $F_X$ the cumulative distribution function.

**Distribution of Smallest Equivalent Ideal.** Consider the random variable $\delta := n/\sqrt{p}$ introduced above, then we we know from Lemma 1 that $\delta \leq 2\sqrt{2}/\pi \approx 0.90$. However, we get a more detailed understanding by studying its probability mass function $f_\delta$, shown in Fig. 2.

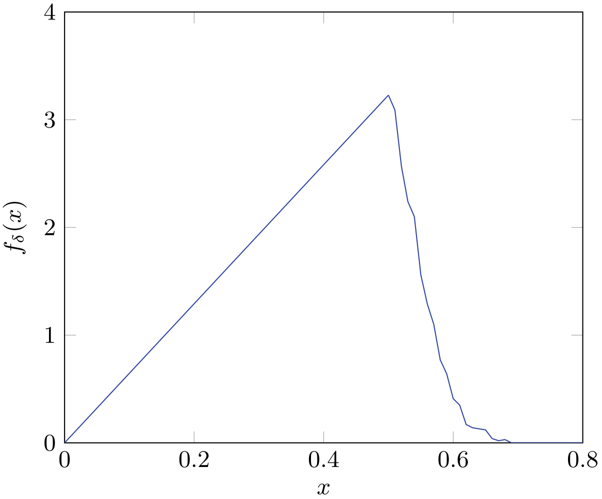

**Fig. 2.** Probability mass function $f_\delta$ for $\delta := n/\sqrt{p}$.

In the SQIsign specification document [2, Section 9.3], a simple counting argument is given that not only explains the shape of $f_\delta$, but also why it is practically independent of $p$. For completeness, we repeat the argument here. The number of left $\mathcal{O}_0$-ideals of given norm $m$ is $\psi(m)$ where $\psi$ is the Dedekind psi function:

$$\psi(m) = m \prod_{\ell \mid m} (1 + 1/\ell) .$$

The number of ideals of norm smaller than a given bound $M$ therefore is

$$\sum_{m \leq M} \psi(m) \simeq \frac{\zeta(2)}{2\zeta(4)} M^2 = \frac{15}{2\pi^2} M^2 .$$

Since there are $\sim p/12$ ideal classes, if we assume that ideals of small norm fall into different classes we would get

$$F_\delta(x) = Pr(\delta < x) \simeq \frac{90}{\pi^2} x^2 .$$

In the order $\mathcal{O}_0$, this assumption is clearly not satisfied, as for example, all the elements of the form $a + b\mathbf{i}$ generate principal ideals, which correspond to a large portion of the small normed ideals, but this does not fundamentally change the quadratic behavior of the cumulative probability distribution for small enough $x$. Our own experiments back up those of [2] and result in the estimate

$$F_\delta(x) = Pr(\delta < x) \simeq \frac{6.46}{2} x^2 .$$

By taking the derivative, we finally obtain that $f_\delta \simeq 6.46 \cdot x$ for $x$ small enough. From Fig. 2 we can see that the linear behavior holds for $x < 0.5$. For $x \geq 0.5$ the dependencies take over, and $f_\delta$ quickly drops down to 0.

**Distribution of 2-Norm of Smallest Solution of $\boldsymbol{ax + by = c \bmod n}$.** Equation (10) is of the form $ax + by = c \bmod n$ where $\gcd(a, n) = 1$ and $n \nmid b$. Let $\epsilon_n := \sqrt{s^2 + t^2}/\sqrt{n}$ denote the random variable defined above, where $(s, t)$ is a solution of smallest norm.

The following example shows that it is impossible to bound $\epsilon_n$ by a constant independent of $n$: assume that $n$ is odd, and set $a = (n-2)$, $b = 2$ and $c = 1$, then the smallest solution is $(s, t) = (1, -(n-3)/2)$, for which $\epsilon_n \simeq \sqrt{n}/2$. However, such examples occur with probability $O(1/n)$, so the probability mass of $f_{\epsilon_n}$ will be concentrated on a finite interval. Furthermore, although the tail of $f_{\epsilon_n}$ clearly depends on $n$, $f_{\epsilon_n}$ quickly converges to some fixed $f_\epsilon$ for $n \to \infty$. Figure 3 plots the probability mass function $f_\epsilon$, Fig. 4 shows the rapid convergence of the $f_{\epsilon_n}$ towards $f_\epsilon$, and finally, Fig. 5 shows the cumulative distribution $F_\epsilon$.

The functions $f_{\epsilon_n}$ were estimated for various $n$, by sampling $10^7$ random values $a, b, c \in (\mathbb{Z}/n\mathbb{Z})^3$, with $a$ invertible, and $b$ non-zero, and computing the shortest solution $s, t$ solving $as + bt \equiv c \pmod{n}$, and evaluating $\epsilon_n = \sqrt{s^2 + t^2}/\sqrt{n}$. As Fig. 4 shows, the functions quickly converge towards $f_\epsilon$, already at $n = 2^{15}$. In practice, $n$ will be much larger than this, thus we conclude that $f_\epsilon$ accurately represents the relevant probability distributions.

**Expected Number of $\boldsymbol{\alpha}$'s.** Now that we have analyzed the probability mass functions $f_\delta$ and $f_\epsilon$, we can finally derive an expression for the expected number of $\alpha$'s required for Qlapoti to terminate. We first list the necessary and sufficient conditions on the tuple $(\alpha, s, t)$.

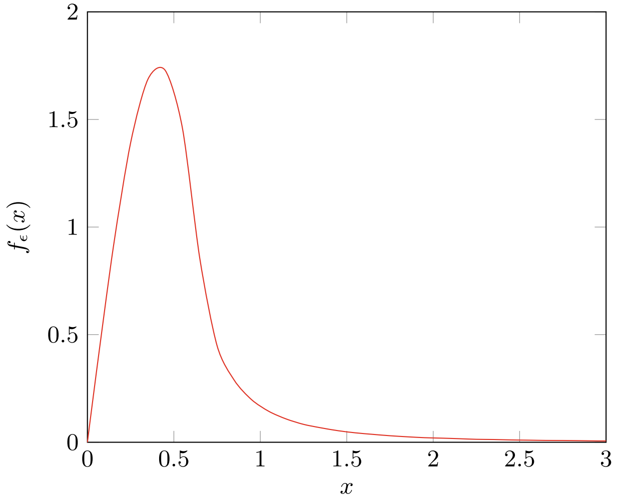

**Fig. 3.** Probability mass function $f_\epsilon$ of $\epsilon_n = \sqrt{s^2+t^2}/\sqrt{n}$ for large $n$.

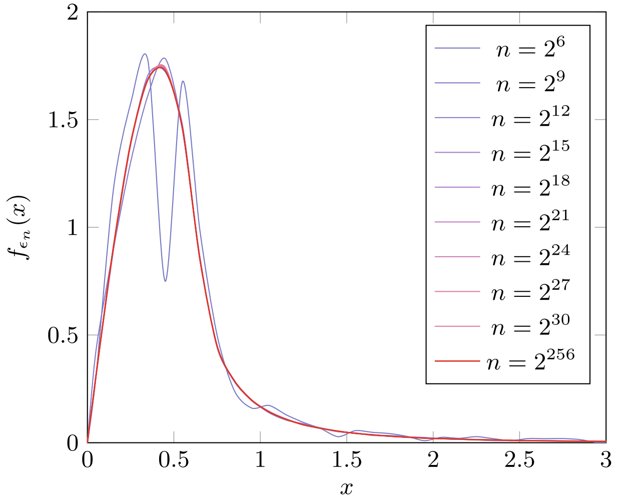

**Fig. 4.** The functions $f_{\epsilon_n}$ for $n = 2^6, 2^9, 2^{12}, \ldots, 2^{30}$, and $n = 2^{256}$, showing the rapid convergence towards $f_\epsilon$.

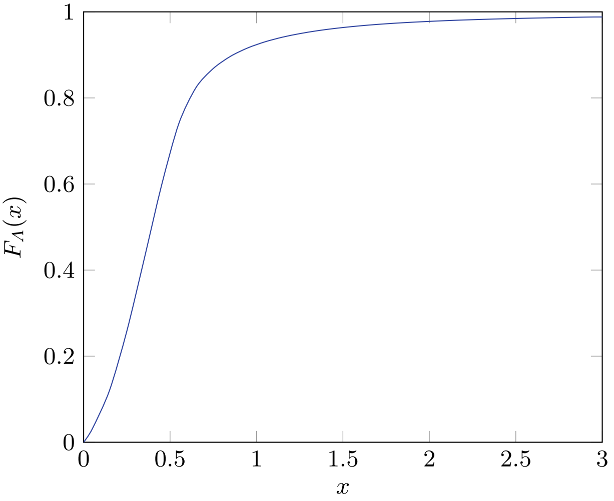

**Fig. 5.** Cumulative distribution function $F_\epsilon$ of $\epsilon_n = \sqrt{s^2+t^2}/\sqrt{n}$ for large $n$.

To obtain small solutions to Eq. (10) we assumed that $\gcd(2a_\alpha, n) = 1$ or that $\gcd(2b_\alpha, n) = 1$ (see check in Step 5). As a first approximation, we assume that $2a_\alpha, 2b_\alpha, n$ are random integers (recall that $a_\alpha, b_\alpha \in \frac{1}{2}\mathbb{Z}$). Then the probability that both pairs are not coprime is $(1 - 6/\pi^2)^2 \simeq 0.15372$, so we succeed with probability $\simeq 0.84628$. Note that the assumption that $n$ behaves as a random integer is not quite correct, as shown in [7]. For example, $n$ will be even with probability 5/8 instead of 1/2. To simplify the analysis we did not take this more complex behaviour into account as it will not fundamentally change our conclusions.

For such $\alpha$, we consider the smallest corresponding $(s, t)$. Equation (9) requires that $w = 2^e - 2r - 2a_\alpha s - 2b_\alpha t > 0$. Since $2^e \sim p/(4f)$, $n \sim \delta\sqrt{p}$ and thus $s, t \sim \sqrt[4]{\delta^2 p}$, this condition is easily satisfied as long as $r \ll p$ and $a_\alpha, b_\alpha \ll p^{3/4}/\sqrt{\delta}$. Note that $a_\alpha, b_\alpha$ can be assumed to be reduced modulo $n$, so the latter condition is always satisfied.

A necessary condition for Eq. (11) to have a solution is that $2w > n(s^2 + t^2)$ (see check in Step 8). As argued before, $w \simeq 2^e - o(2^e)$, so it suffices that

$$2^{e+1} > \delta^2 \epsilon^2 p,$$

where we used that $\epsilon = \sqrt{s^2+t^2}/\sqrt{n}$ and $\delta = n/\sqrt{p}$. Recall that $p = f2^{e+2}-1$ for some small cofactor $f$, then a necessary condition for the algorithm to terminate is

$$2f < \delta^{-2}\epsilon^{-2}\,.$$

Since $\delta$ is fixed for a given input ideal, the smallest solution $(s, t)$ has to satisfy

$$\epsilon < 1/\delta\sqrt{2f}\,.$$

Denote $u := 1/\delta\sqrt{2f}$. Since we know the cumulative distribution $F_\epsilon$, the number of $\alpha$'s to try is explicitly given by $1/F_\epsilon(u)$.

Using the worst-case bound for $\delta_{wc} = 2\sqrt{2}/\pi$ and the average-case bound $\delta_{ac} = 0.37$, we can explicitly compute the number of $\alpha$'s required for a small solution $(s,t)$ to exist and for Eq. 11 to have positive RHS. We call such $(\alpha, s, t)$ tuple *good*. We compute this for the three SQIsign parameter sets in Table 1.

**Table 1.** Required number of good $(\alpha, s, t)$ tuples for SQIsign parameter sets.

| NIST Level | $p$ | $e$ | $u_{wc}$ | $\#\alpha$ (wc) | $u_{ac}$ | $\#\alpha$ (ac) |
|---|---|---|---|---|---|---|
| I | $5 \cdot 2^{248} - 1$ | 246 | 0.351 | 3.23 | 0.851 | 1.18 |
| III | $65 \cdot 2^{376} - 1$ | 374 | 0.097 | 37.7 | 0.236 | 6.66 |
| V | $27 \cdot 2^{500} - 1$ | 498 | 0.151 | 15.7 | 0.366 | 3 |

Now that we have $\alpha$ and $(s,t)$ such that Eq. (11) potentially has a solution, we still need to find it. First, this requires that $z$ can be written as a sum of two squares. By a theorem of Landau [21], the number of positive integers smaller than some bound $B$ that can be written as a sum of two squares is asymptotically

$$\frac{\lambda B}{\sqrt{\log(B)}},$$

with $\lambda \simeq 0.76422$ the Landau–Ramanujan constant. Note that $z$ is bounded by $B = 2^{e+1}/\delta\sqrt{p} \sim \sqrt{p}/(2f\delta)$. Taking the extra modulo 4 restrictions on $z, s, t$ into account, which hold with probability 1/4, we conclude we need to try on average

$$5.23\sqrt{\log(\sqrt{p}/(2f\delta))} \tag{12}$$

good $(\alpha, s, t)$ tuples before finding a solution. The above analysis assumes we would compute a full factorization of $z$, which is not the case in practice. In particular, we use trial division with a bound up to 500, and then test the reminder for primality. We can derive a worst case bound by simply only considering primes $z = 1 \bmod 4$. This would result in a worst case bound of

$$8\log(\sqrt{p}/(2f\delta)) \tag{13}$$

good $(\alpha, s, t)$ tuples to try. Combining these results with Table 1, we finally obtain the range for the overall number of $\alpha$'s to try. In Table 2 we list the worst-case (wc) scenario, where we combine the worst case from Table 1 with Eq. (13) and the optimal case (oc) where we combine the average case from Table 1 with Eq. (12). Note that these (wc) numbers give an upper bound on the average number of $\alpha$'s we need to try to succeed.

**Table 2.** Average number of $\alpha$'s required for Qlapoti to terminate.

| NIST Level | $p$ | $e$ | $\#\alpha$ (wc) | $\#\alpha$ (oc) |
|---|---|---|---|---|
| I | $5 \cdot 2^{248} - 1$ | 246 | 2185 | 57 |
| III | $65 \cdot 2^{376} - 1$ | 374 | 38495 | 395 |
| V | $27 \cdot 2^{500} - 1$ | 498 | 21484 | 206 |

To simplify the implementation of the 2D-step, one requires that $I_1$ and $I_2$ are odd norm and coprime. If one takes this restriction into account, the numbers in Table 2 have to be multiplied by $\pi^2/2 \sim 4.93$.

**Failure Probability for SQIsign Parameters.** Finally, we count the number of choices of generators $\alpha$ that we have available for a fixed instance, to see that it is exponential in $\log n$, and derive the final failure probabilities for the SQIsign parameters. We employ the fact that $\mathcal{O}/n\mathcal{O} \simeq M_2(\mathbb{Z}/n\mathbb{Z})$, and thus $I/n\mathcal{O}$ can be identified with a (necessarily) principal, non-trivial left ideal in $M_2(\mathbb{Z}/n\mathbb{Z})$. By [19, Lemma 7.2], $I/n\mathcal{O}$ can be generated by an element of the form $\begin{pmatrix} x & y \\ 0 & 0 \end{pmatrix}$. Furthermore, like for all principal ideals, all other generators are obtained by acting with $M_2(\mathbb{Z}/n\mathbb{Z})^\times = GL_2(\mathbb{Z}/n\mathbb{Z})$. Since

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x & y \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} ax & ay \\ cx & cy \end{pmatrix},$$

we see that the generators of $I/n\mathcal{O}$ are in bijection with pairs $a, c \in \mathbb{Z}/n\mathbb{Z}$ such that $\gcd(a, c, n) = 1$ (it is easy to see that this is a necessary and sufficient condition to find $b, d$ such that $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in GL_2(\mathbb{Z}/n\mathbb{Z})$). The number of such pairs is

$$n^2 \prod_{p|n} \left(1 - \frac{1}{p^2}\right) > n^2 \prod_{p \text{ prime}} \left(1 - \frac{1}{p^2}\right) = n^2 \frac{1}{\zeta(2)} = n^2 \frac{6}{\pi^2} \approx 0.607927 n^2,$$

where $\zeta(s)$ denotes the Riemann zeta function, and $\zeta(2) = \frac{\pi^2}{6}$ is famously classical.

The norm condition on the lift of these generators implies that the coefficients $c_\alpha, d_\alpha$ should satisfy

$$2^e > 2\mathrm{nrd}(\alpha)/n \approx 2 \cdot p(c_\alpha^2 + d_\alpha^2)/n,$$

and thus it suffices that

$$|c_\alpha|, |d_\alpha| < \sqrt{n}\sqrt{\frac{2^e}{4p}}$$

Assuming these behave as random integers between 0 and $n$, this happens with probability

$$\left(\frac{\sqrt{n}}{n}\sqrt{\frac{2^e}{4p}}\right)^2 = \frac{2^e}{4pn}.$$

Thus the expected number of suitable $\alpha$'s is $\approx 0.607927n \cdot \frac{2^e}{4p}$.

We now compute the minimal $n$ such that Qlapoti fails with probability less than $2^{-(\lambda+1)}$. Here we make the assumption that the probability of success per $\alpha$ is the same for all $\alpha$. The probability for Qlapoti to terminate therefore follows a geometric distribution. Let $c$ denote the mean of this distribution, i.e. the average number of $\alpha$'s required to terminate, then the probability of success for a single $\alpha$ is $q = 1/c$. The probability that Qlapoti fails to terminate within $k$ steps, i.e. $k$ $\alpha$'s, is $(1-q)^k$, and we thus require $(1-q)^k < 2^{-(\lambda+1)}$. Taking log's and using $\log(1-x) \simeq -x$ for small $x$ gives $k > (\lambda+1)\log(2)c$. Therefore Qlapoti is guaranteed to terminate for all $n$ that satisfy

$$n > (\lambda+1) \cdot \frac{4\log(2) \cdot p \cdot c}{0.607927 \cdot 2^e}.$$

Let $L$ denote the right hand side of the above equation. Then as a very crude estimate, we assume that Qlapoti always fails whenever $n \leq L$, and it fails with probability $< 2^{-(\lambda+1)}$ for $n > L$, so we obtain:

$$P(\mathsf{Qlapoti}\text{ fails}) < P(n \leq L) + P(\mathsf{Qlapoti}\text{ fails} \mid n > L) < F_\delta(L/\sqrt{p}) + 2^{-(\lambda+1)},$$

As before, $F_\delta(L/\sqrt{p}) \approx \frac{6.46}{2}(L/\sqrt{p})^2$, and thus we obtain a lower bound on the failure rate by finding the largest $\lambda$ such that

$$2^{-(\lambda+1)} > \frac{6.46}{2}(L/\sqrt{p})^2 = \frac{6.46}{2} \cdot \left((\lambda+1) \cdot \frac{4\log(2) \cdot \sqrt{p} \cdot c}{0.607927 \cdot 2^e}\right)^2.$$

as this gives $P(\mathsf{Clapoti}\text{ fails}) < 2^{-(\lambda+1)} + 2^{-(\lambda+1)} = 2^{-\lambda}$. Specifically, for the SQIsign parameter sets, we can read off $c$ from Table 1, and obtain the final failure probabilities, which are confidently within the security level (see Table 3).

**Table 3.** The final upper bound of the failure rate of Qlapoti applied to the SQIsign parameters.

| NIST Level | $p$ | $c$ | $e$ | upper bound on failure rate |
|---|---|---|---|---|
| I | $2^{248} \cdot 5 - 1$ | 2185 | 246 | $2^{-197}$ |
| III | $2^{376} \cdot 65 - 1$ | 38495 | 374 | $2^{-312}$ |
| V | $2^{500} \cdot 27 - 1$ | 21484 | 498 | $2^{-438}$ |

*Remark 1.* We also remark that the failure rate can even be further improved by trying more $(s,t)$ pairs. Indeed, the algorithm fails only when $n$ is tiny, in which case there is no restriction on $(s,t)$ being "short" (relative to $n$) anymore, and one can rerandomize by trying new $(s,t)$ pairs instead. Thus, we can essentially get an algorithm that always works, as long as there is a single $\alpha$ that is small enough. However, given that the simpler algorithm already has completely negligible failure probability for practical purposes, we decided to stick with the simpler description of our algorithm.

## 5 Implementation and Results

In this section we discuss some implementation details and optimisations, before presenting our results. Our implementation is available at

https://github.com/KULeuven-COSIC/Qlapoti

### 5.1 Reusing Lattice Reductions

In our implementation, Algorithm 2 is implemented exactly as described, except for the small practical improvement that we rerandomize $\alpha$ by trying new $\lambda\alpha$ for small integers $\lambda$ satisfying $\gcd(\lambda, n) = 1$. This has the advantage of being cheaper to find short $s, t$ pairs (Line 6), since the CVP instance will be for the same lattice (but with a new target vector), thus saving one lattice reduction and matrix inversion per iteration.

### 5.2 Non-gluing Isogenies

Recall that after Algorithm 2 outputs a solution, $\mathsf{IdealToIsogeny}$ proceeds by deriving the kernel of the corresponding isogeny $\Phi : E_0 \times E_0 \to E_J \times E'$ to be

$$\ker \Phi = \langle ([d_1]P_0, \theta(P_0)), ([d_1]Q_0, \theta(Q_0)) \rangle \text{ for } E_0[2^e] = \langle P_0, Q_0 \rangle.$$

**Endomorphisms.** According to [18, Thm. 3], the first $(2,2)$-isogeny will land on a product of elliptc curves (instead of gluing into a Jacobian) if and only if its kernel is the graph of an automorphism, on $E_0$ restricted to the 2-torsion. Since $d_1$ is odd, in our case the kernel is of the form $\{(P, \theta(P)) \mid P \in E_0[2]\}$. To land on a product, we thus have only two possibilities, namely $\theta(P) = P$ and $\theta(P) = \mathbf{i}P$ for $P \in E_0[2]$.

Expressing $\theta$ in the $\mathbb{Z}$-basis for $\mathcal{O}_0$ we have $\theta = x_1 + x_2\mathbf{i} + x_3\frac{\mathbf{i}+\mathbf{j}}{2} + x_4\frac{1+\mathbf{k}}{2}$, for some $x_i \in \mathbb{Z}$. We note that if either $x_3$ or $x_4$ is odd, neither of these two cases can happen. Thus, we restrict to the case $\theta = x_1 + x_2\mathbf{i} + x_3\mathbf{j} + x_4\mathbf{k}$, with $x_i \in \mathbb{Z}$. Moreover, both $\mathbf{i} + \mathbf{j}$ and $1 + \mathbf{k}$ are trivial on the 2-torsion. The action of $\theta$ on $E_0[2]$ can therefore be described by

$$\theta \bmod 2 = (x_1 + x_4 \bmod 2) + (x_2 + x_3 \bmod 2)\mathbf{i}.$$

The non-gluing cases will be the ones in which exactly one of the two coefficients is 1 (note that they cannot be both 0, since otherwise $\theta$ would have even norm). For instance, if $x_1 + x_4 \bmod 2 = 1$ (and thus $x_2 + x_3 \bmod 2 = 0$ by assumption), then $\theta$ acts as the identity on $E_0[2]$.

If we are in one of those two cases, to compute the first step we need to determine the $(2, 2)$-isogeny $\rho$ whose kernel coincides with the first step of the kernel of $\Phi$. Following the discussion above, it is immediate that if $x_1 + x_4 \equiv 1 \bmod 2$ then we have $\ker(\rho) = \{(P, P) | P \in E_0[2]\}$; if otherwise $x_2 + x_3 \equiv 1 \bmod 2$ then $\ker(\rho) = \{(P, \mathbf{i}P) | P \in E_0[2]\}$. From this we deduce

$$\rho((P, Q)) := \begin{cases} (P + Q, P - Q), & \text{if } x_1 + x_4 \equiv 1 \bmod 2 \\ (\mathbf{i}(P) + Q, \mathbf{i}(P) - Q), & \text{if } x_2 + x_3 \equiv 1 \bmod 2 \end{cases}.$$

In particular, after applying $\rho$ the kernel of the remaining steps will be

$$K_1 := \langle (P_1, P_2), (Q_1, Q_2)\rangle := \langle (\theta^+(P_0), \theta^-(P_0)), (\theta^+(Q_0), \theta^-(Q_0))\rangle,$$

where

$$\theta^\pm := \begin{cases} d_1 \pm \theta, & \text{if } x_1 + x_4 \equiv 1 \bmod 2 \\ d_1\mathbf{i} \pm \theta, & \text{if } x_2 + x_3 \equiv 1 \bmod 2 \end{cases}.$$

Notice that since the $\theta^\pm$ are divisible by 2, the new kernel indeed has the correct order $2^{e-1}$.

The next question is whether this phenomenon can repeat in the following steps. We restrict to the case $x_1 + x_4 \equiv 1 \bmod 2$; the other case follows similarly. In this case, the kernel of the next $(2, 2)$-isogeny step is given by

$$[2^{e-2}]K_1 = \left\{ \left( \left(\frac{\theta^+}{2}\right) P, \left(\frac{\theta^-}{2}\right) P \right) \mid P \in E_0[2] \right\}.$$

To deduce whether this is again the graph of an automorphism, we need to check whether $[2^{e-2}]K_1$ is in the form $\{(P, P) \mid P \in E_0[2]\}$ or $\{(P, \mathbf{i}P) \mid P \in E_0[2]\}$.

To do so, we first express both $\theta^\pm/2$ in the standard $\mathbb{Z}$-basis for $\mathcal{O}_0$. We first note that the constant term is $\frac{d_1 \mp (x_1 - x_4)}{2}$. These two quantities will differ modulo 2 since $x_1 - x_4$ is odd, and as such the kernel cannot be the graph of the identity. To see why this kernel cannot be the graph of $\mathbf{i}$, we observe that applying $\mathbf{i}$ on the 2-torsion simply permutes the coefficients. Conversely, the coefficients of $\theta^+/2$ and $\theta^-/2$, when considered modulo 2, differ in exactly one component. Therefore, we rule out this case as well, and after the first step we cannot have another endomorphism.

**Diagonal Isogenies.** After this first step however there may still be diagonal steps. Namely, the $i$-th step can be of the form $\Phi_i = (\varphi_i^{(1)}, \varphi_i^{(2)})$, where $\varphi_i^{(1)}$ and $\varphi_i^{(2)}$ are one dimensional 2-isogenies.

This will happen if $(P, 0) \in \ker(\Phi_i)$, or equivalently if the norm of $\theta^+/2$ has non-trivial 2-valuation. Note that as already observed, this can never be the case with $\theta$ itself, further showing that the first step cannot be diagonal.

*Number of Diagonal Steps.* We first detect the number of diagonal steps we take after the first step (before gluing to a Jacobian). Since the kernel of each step must be isotropic, the kernel of a diagonal step will be of the form

$$\{(P, 0), (0, Q), (P, Q), (0, 0)\}.$$

for 2-torsion points $P$ and $Q$. As a result, the 2-valuation of norm of $\theta^+/2$ is the same as that of $\theta^-/2$. Furthermore, this shows that $\theta^+/2$ cannot factor through multiplication by 2, as otherwise $P$ would be in the kernel of $\theta^+/2$ as well. The number of diagonal steps will then be exactly the 2-valuation of $\mathrm{nrd}(\theta^+/2)$.

*The Shape of the Kernel.* Next, we determine the kernel of the diagonal steps. A key observation is that we only need to determine the shape of the kernel once after the first step. Indeed, let $v = v_2(\mathrm{nrd}(\theta^+/2))$ be the 2-valuation, and set $D = 2^{e-v-1}$. The kernel of the diagonal steps will be

$$[D]K_1 = \langle ([D]P_1, [D]P_2), ([D]Q_1, [D]Q_2) \rangle$$

Since this kernel is isotropic, it must be isomorphic to $\mathbb{Z}_{2^v} \times \mathbb{Z}_{2^v}$, and since the order of all elements divides $2^v$ one among $[D]P_1$ and $[D]Q_1$ and one among $[D]P_2$ and $[D]Q_2$ respectively must have order $2^v$.

Assume $[D]P_1$ has full order. As the point $([2^{e-2}]P_1, 0)$ must be in the kernel, $[D]P_2$ cannot have full order. Therefore, for a diagonal step $\Phi_i = (\varphi_i^{(1)}, \varphi_i^{(2)})$, we have that

$$\ker(\varphi_i^{(1)}) := \langle [D]P_1 \rangle \text{ or } \langle [D]Q_2 \rangle,$$
$$\ker(\varphi_i^{(2)}) := \langle [D]P_2 \rangle \text{ or } \langle [D]Q_1 \rangle.$$

One way to detect which of the cases we are in is, for instance, to check the order of the pairing $e_{2^e}(P_1, Q_2)$. Note that this can be expressed as a linear combination of $e_{2^e}(P_0, Q_0)$, $e_{2^e}(P_0, \mathbf{i}Q_0)$, and so on. All these quantities can be precomputed, making this method quite efficient. Another solution would be to precompute the three norm-2 left ideals in $\mathcal{O}_0$ and the corresponding kernels on the 2-torsion, and check which one corresponds to $\mathcal{O}_0(\theta^+/2) + \mathcal{O}_0 2$ and $\mathcal{O}_0(\theta^-/2) + \mathcal{O}_0 2$.

### 5.3 Results

We now present the results of our implementation of Qlapoti in both SageMath and C. We first present statistics on the number of rerandomizations, and see that it matches well with the theoretical results obtained in Sect. 4. We then present the benchmarks on performance, both related to timings and memory usage.

**Statistics.** We ran our implementation of Qlapoti 10,000 times for each security level and measured the number of rerandomizations needed, to compare against Table 2. The results are presented in Table 4.

**Table 4.** The predicted number of rerandomizations (taken from Table 2), compared to the observed average number of rerandomizations. The observed numbers were measured over 10,000 runs.

| NIST Level | predicted #$\alpha$ (wc) | predicted #$\alpha$ (oc) | avg. #$\alpha$ |
|---|---|---|---|
| I | 2185 | 57 | 65.45 |
| III | 38495 | 395 | 569.2 |
| V | 21484 | 206 | 351.6 |

On average, our random instances behave closely to the predicted optimal-case assumptions. This adds confidence to the already negligible failure rate reported in Sect. 4, which was computed under the predicted worst-case assumptions. Further, we used our data to verify the assumption that the success probability behaves independently on the sampled $\alpha$'s (see Fig. 6).

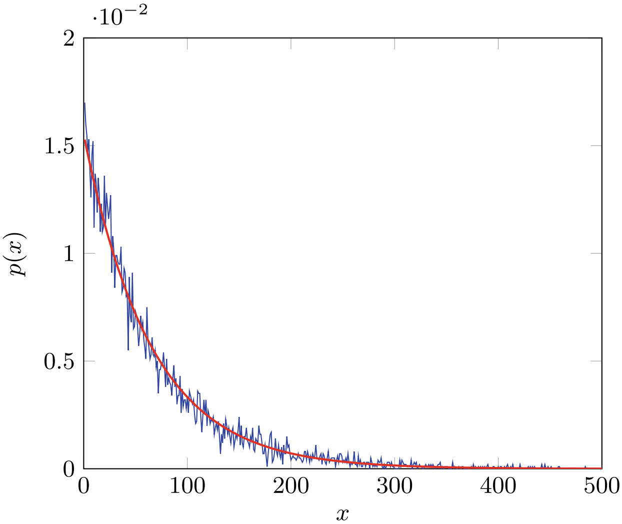

**Fig. 6.** The observed probabilty $p(x)$ of requiring exactly $x$ values of $\alpha$ for Qlapoti to terminate over 10,000 runs (for NIST Level 1) in blue, compared to a geometric distribution given by the mean in red. (Color figure online)

**Timings and Measurements.** With our SageMath implementation, we compare directly with the SageMath implementation of IdealToIsogeny from [5]. The results were obtained by measuring the average over 500 runs, on an Intel Core Ultra 7 165H. The results can be found in Table 5.

**Table 5.** Timings comparing IdealToIsogeny using the technique currently used in SQIsign and the one presented in this work, given in wall-clock time. The final column represents the improvement factor.

| NIST level | Previous work [5] | This work | Improvement |
|---|---|---|---|
| I | 0.434s | 0.166s | x2.6 |
| III | 0.849s | 0.386s | x2.2 |
| V | 1.143s | 0.490s | x2.3 |

We also give results for our implementation of Qlapoti in `C`, using the NIST round 2 implementation of SQIsign [2]. Before discussing the impact of Qlapoti on SQIsign, we point out that for the `C` implementation, we restrict Qlapoti to only output isogenies that do not contain any diagonal isogenies, thus we also give new SageMath timings with this restriction. This is because the $x$-only arithmetic creates additional complications when evaluating an embedded isogeny that contains diagonal steps. This in turn leads to extra rerandomizations, and the improvement factor becomes lower, as seen in Table 6.

Conceptually, the limitation of restricted Qlapoti outputs can be easily overcome by implementing $xy$-arithmetic for the diagonal steps. However, this would be the only place in the SQIsign-implementation where $xy$-arithmetic is needed, thus leading to a lot of extra code complexity for a (potentially) small gain. We leave it as future work to optimize either the potential diagonal steps, or improving Qlapoti with restricted outputs.

**Table 6.** Timings comparing IdealToIsogeny in SageMath, when restricting Qlapoti to not output diagonal isogenies. Compared to Table 5, we see that the improvement factor is lower with this restriction.

| NIST level | Previous work [5] | This work | Improvement |
|---|---|---|---|
| I | 0.434s | 0.171s | x2.5 |
| III | 0.849s | 0.446s | x1.9 |
| V | 1.143s | 0.515s | x2.2 |

In Table 7, we give benchmarks which measure the improvement gained for the key generation and signing procedure of SQIsign and PRISM. The SageMath timings are given using the same setup as above. The `C` implementation timings are given over 10,000 runs. It was compiled with `gcc`, with the `CMAKE` options `-DSQISIGN_BUILD_TYPE=ref -DCMAKE_BUILD_TYPE=Release` on an Intel i7-11850H processor with turboboost disabled.

We note that the relative improvement IdealToIsogeny is currently lower in `C` than in SageMath. We explain this observation by noting that, in `C`, the quaternion operations are more costly (relative to the finite field operations) than in SageMath, since those cannot benefit from the optimized field arithmetic, and instead uses unbounded integers. This is especially noticable in Level III and V, which has the biggest cofactor (ref. Table 3), which negatively impacts the running time of Qlapoti. Thus, these higher levels would benefit the most from allowing the earlier mentioned diagonal steps, as it effectively halves the expected running time of Qlapoti.

**Table 7.** Benchmarks to measure the impact of Qlapoti on the signature schemes SQIsign and PRISM. The comparison with PRISM is in SageMath, with the implementation from [5], while the comparison with SQIsign is in C, compared against the NIST round 2 submission [2]. The timings in SageMath are given in wall-clock time, while the measurements in C are given in Megacycles.

| Protocol | Algorithm | Previous work | This work | Improvement |
|---|---|---|---|---|
| SQIsign-LVLI | KeyGen | 123 Mcy | 67.5 Mcy | x1.8 |
| | Signing | 282 Mcy | 172 Mcy | x1.6 |
| SQIsign-LVL3 | KeyGen | 315 Mcy | 287 Mcy | x1.1 |
| | Signing | 719 Mcy | 667 Mcy | x1.1 |
| SQIsign-LVL5 | KeyGen | 516 Mcy | 355 Mcy | x1.5 |
| | Signing | 1218 Mcy | 900 Mcy | x1.4 |
| PRISM-LVLI | KeyGen | $0.484s$ | 0.252s | x1.9 |
| | Signing | 0.593s | 0.322s | x1.7 |
| PRISM-LVL3 | KeyGen | 0.915s | 0.544s | x1.7 |
| | Signing | 1.328s | 0.808s | x1.6 |
| PRISM-LVL5 | KeyGen | 1.436s | 0.758s | x1.9 |
| | Signing | 2.017s | 1.426s | x1.4 |

We also present a comparison of the memory consumption. This metric is important when considering execution on constrained devices. Despite SQIsign's large runtimes, there has been recent work on porting it to smaller CPUs, such as the Cortex-M4 [1]. However, signing proved to be less portable than verification, due not only to its slowness, but also to its memory management. In particular, the large amount of dynamically allocated memory (mostly through the GMP library), as well as the overall memory usage, presents a challenge for portability [1, Remark 5].

As previously discussed, the large memory footprint in the SQIsign NIST submission (round 2) is mostly due to the previous IdealToIsogeny algorithm, which needed to compute large tables of short elements in 7 ideals (or 8 at level 3). In Table 8 we present a comparison of the heap memory usage when using Qlapoti instead.

**Table 8.** Heap usage by a reference/Release build of the SQIsign NIST2 implementation with and without Qlapoti, as measured by valgrind's massif tool. Average over 100 runs on same machine as C code benchmarks. Measures were taken with the `sqisign_test_scheme_lvl[x]` executable for Level `x`.

| NIST Level | Previous work [2] | This work | Improvement |
|---|---|---|---|
| I | 0.42 MiB | 38 KiB | x11 |
| III | 1.9 MiB | 56 KiB | x34 |
| V | 1.7 MiB | 74 KiB | x23 |

Clearly, Qlapoti improves the memory usage significantly, particularly at higher security levels, and therefore gives hope that SQIsign can also be executed on smaller devices in the future, provided that the use of dynamic memory allocation can be avoided.

Another interesting fact we have observed is that the size of the executable containing SQIsign also decreases when using Qlapoti in the ideal-to-isogeny computation of SQIsign (see Table 9). Again, this is related to the fact that the precomputation data related to the rerandomization procedure currently takes up a significant portion of the size of the executable.

**Table 9.** Size of the `sqisign_test_scheme_lvl[x]` executable, where `x` is as in the first column, for a ref/Release build on an intel i5-6300U, compiled with `gcc` 13.3.

| NIST Level | Previous work [2] | This work |
|---|---|---|
| I | 254 KiB | 212 KiB |
| III | 278 KiB | 221 KiB |
| V | 282 KiB | 233 KiB |

**Acknowledgements.** The work of Giacomo Borin and Sina Schaeffler was supported in part by SNSF Consolidator Grant CryptonIs 213766. Maria Corte-Real Santos was supported by the European Research Council under grant No. 101116169 (AGATHA CRYPTY). Riccardo Invernizzi was supported by Fonds voor Wetenschappelijk Onderzoek (FWO) with reference number 1138925N. Jonathan Komada Eriksen, Riccardo Invernizzi and Frederik Vercauteren were supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement ISOCRYPT - No. 101020788), by the Research Council KU Leuven grant C14/24/099 and by CyberSecurity Research Flanders with reference number VOEWICS02.

## References

[1] Aardal, M.A., et al.: Optimized One-Dimensional SQIsign Verification on Intel and Cortex-M4. Cryptology ePrint Archive, Report 2024/1563 (2024). https://eprint.iacr.org/2024/1563

[2] Aardal, M.A., et al.: SQISign Specification. Technical report. version 2.0 from 2025-02-05. National Institute of Standards and Technology (2025). https://sqisign.org

[3] Aardal, M.A., Basso, A., De Feo, L., Patranabis, S., Wesolowski, B.: A Complete Security Proof of SQIsign. Cryptology ePrint Archive, Paper 2025/379 (2025). https://eprint.iacr.org/2025/379

[4] Aono, Y., Espitau, T., Nguyen, P.Q.: Random Lattices: Theory and Practice (2018). https://espitau.github.io/bin/random_lattice.pdf

[5] Basso, A., et al.: PRISM: simple and compact identification and signatures from large prime degree isogenies. In: Jager, T., Pan, J. (eds.) Public- Key Cryptography – PKC 2025, pp. 300–332. Springer, Cham (2025). https://doi.org/10.1007/978-3-031-91826-1_10

[6] Basso, A., et al.: SQIsign2DWest - the fast, the small, and the safer. In: Chung, K.M., Sasaki, Y. (eds.) Advances in Cryptology – ASIACRYPT 2024, Part III. LNCS, Kolkata, India, vol. 15486, pp. 339–370. Springer, Singapore (2024). https://doi.org/10.1007/978-981-96-0891-1_11

[7] Castryck, W., Chen, M., Invernizzi, R., Lorenzon, G., Vercauteren, F.: Breaking and Repairing SQIsign2D-East. Cryptology ePrint Archive, Paper 2024/1453 (2024). https://eprint.iacr.org/2024/1453

[8] Castryck, W., Decru, T.: An efficient key recovery attack on SIDH. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023, Part V. LNCS, Lyon, France, vol. 14008, pp. 423-447. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30589-4_15

[9] Cornacchia, G.: Su di un metodo per la risoluzione in numeri interi dell'equazione $\sum_{h=0}^{n} c_h x^{n-h} y^h = p$. Battaglini 46, 33–90 (1908)

[10] Dartois, P., et al.: PEGASIS: Practical Effective Class Group Action using 4-Dimensional Isogenies. Cryptology ePrint Archive, Paper 2025/401 (2025). https://eprint.iacr.org/2025/401

[11] Dartois, P., Leroux, A., Robert, D., Wesolowski, B.: SQIsignHD: new dimensions in cryptography. In: Joye, M., Leander, G. (eds.) Advances in Cryptology – EUROCRYPT 2024, Part I. LNCS, Zurich, Switzerland, vol. 14651, pp. 3–32. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-58716-0_1

[12] De Feo, L.: Mathematics of isogeny based cryptography. arXiv preprint arXiv:1711.04062 (2017)

[13] De Feo, L., Kohel, D., Leroux, A., Petit, C., Wesolowski, B.: SQISign: compact post-quantum signatures from quaternions and isogenies. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020. LNCS, vol. 12491, pp. 64–93. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64837-4_3

[14] De Feo, L., Leroux, A., Longa, P., Wesolowski, B.: New algorithms for the Deuring correspondence – towards practical and secure SQIsign signatures. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023, Part V. LNCS, Lyon, France, vol. 14008, pp. 659–690. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30589-4_23

[15] Deuring, M.: Die Typen der Multiplikatorenringe elliptischer Funktionenkörper. Abhandlungen aus dem mathematischen Seminar der Universität Hamburg, vol. 14, pp. 197–272. Springer, Heidelberg (1941)

[16] Galbraith, S.D., Petit, C., Silva, J.: Identification protocols and signature schemes based on supersingular isogeny problems. In: Takagi, T., Peyrin, T. (eds.) Advances in Cryptology – ASIACRYPT 2017, Part I. LNCS, Hong Kong, China, vol. 10624, pp. 3–33. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70694-8_1

[17] Hülsing, A., et al.: SPHINCS+. Technical report. National Institute of Standards and Technology (2022). https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022

[18] Kani, E.: The existence of curves of genus two with elliptic differentials. J. Numb. Theory **64**(1), 130–161 (1997). ISSN: 0022- 314X, 1096-1658. https://doi.org/10.1006/jnth.1997.2105

[19] Kirschmer, M., Voight, J.: Algorithmic enumeration of ideal classes for quaternion orders. SIAM J. Comput. **39**(5), 1714–1747 (2010)

[20] Kohel, D., Lauter, K., Petit, C., Tignol, J.P.: On the quaternion-isogeny path problem. LMS J. Comput. Math. **17**, 418–432 (2014)

[21] Landau, E.: Über die Einteilung der positiven ganzen Zahlen in vier Klassen nach der Mindestzahl der zu ihrer additiven Zusammensetzung erforderlichen Quadrate. Archiv der Mathematik und Physik **13**(3), 305–312 (1908)

[22] Lyubashevsky, V., et al.: Crystals-dilithium. Technical report. National Institute of Standards and Technology (2022). https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022

[23] Maino, L., Martindale, C., Panny, L., Pope, G., Wesolowski, B.: A direct key recovery attack on SIDH. In: Hazay, C., Stam, M. Advances in Cryptology – EUROCRYPT 2023, Part V. LNCS, Lyon, France, vol. 14008, pp. 448–471. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30589-4_16

[24] Morain, F., Nicolas, J.L.: On Cornacchia's algorithm for solving the diophantine equation $u^2 + dv^2 = m$". Projet **1000**, a1 (1990)

[25] Nakagawa, K., Onuki, H.: QFESTA: efficient algorithms and parameters for FESTA using quaternion algebras. In: Reyzin, L., Stebila, D. (eds.) Advances in Cryptology – CRYPTO 2024, Part V. LNCS, Santa Barbara, CA, USA, vol. 14924, pp. 75–106. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-68388-6_4

[26] Nakagawa, K., et al.: SQIsign2D-east: a new signature scheme using 2-dimensional isogenies. In: Chung, K.M., Sasaki, Y. Advances in Cryptology – ASIACRYPT 2024, Part III. LNCS, Kolkata, India, vol. 15486, pp. 272–303. Springer, Singapore (2024). https://doi.org/10.1007/978-981-96-0891-1_9

[27] NIST. Post-Quantum Cryptography: Additional Digital Signature Schemes (2022). https://csrc.nist.gov/projects/pqc-dig-sig

[28] Page, A., Robert, D.: Introducing clapoti(s): Evaluating the isogeny class group action in polynomial time. Cryptology ePrint Archive, Report 2023/1766 (2023). https://eprint.iacr.org/2023/1766

[29] Page, A., Wesolowski, B.: The supersingular endomorphism ring and one endomorphism problems are equivalent. In: Joye, M., Leander, G. (eds.) Advances in Cryptology – EUROCRYPT 2024, Part VI. LNCS, Zurich, Switzerland, vol. 14656, pp. 388–417. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-58751-1_14

[30] PQ-SORT: Post-Quantum Signatures On-Ramp Tests. https://pqsort.tii.ae/. Accessed 14 May 2025

[31] Prest, T., et al.: FALCON. Technical report. National Institute of Standards and Technology (2022). https://csrc.nist.gov/Projects/post-quantum-cryptography/selectedalgorithms-2022

[32] Robert, D.: Breaking SIDH in polynomial time. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023, Part V. LNCS, Lyon, France, vol. 14008, pp. 472–503. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30589-4_17

[33] Robert, D.: On the efficient representation of isogenies (a survey). Cryptology ePrint Archive, Report 2024/1071 (2024). https://eprint.iacr.org/2024/1071

[34] Schwabe, P., et al.: Crystals-kyber. Technical report. National Institute of Standards and Technology (2022). https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022

[35] Silverman, J.H.: The Arithmetic of Elliptic Curves, vol. 106. Springer, Heidelberg (2009)

[36] Voight, J.: Quaternion Algebras, vol. 288, pp. xxiii+885. Graduate Texts in Mathematics. Springer, Cham (2021). ISBN: 978-3-030-56692-0; 978-3-030-56694-4. https://doi.org/10.1007/978-3-030-56694-4