Qlapoty: Improved analysis and efficiency for quaternionic ideal to isogeny transformation

Max Duparc[1], Antonin Leroux[2,3], Sina Schaeffler[4,5]

[1] EPFL, Lausanne, Switzerland
[2] DGA-MI, Bruz, France
[3] IRMAR - UMR 6625, Université de Rennes, France
[4] ETH Zurich, Zurich, Switzerland
[5] IBM Research Europe, Zurich, Switzerland

Abstract. The quaternionic ideal-to-isogeny translation is a central building block of SQIsign. While the Qlapoti algorithm by Borin, Invernizzi, Corte-Real Santos, Eriksen, Mula, Schaeffler and Vercauteren significantly simplified and accelerated this step, it does not treat several technical details in sufficient depth, resulting in a flawed analysis of its failure probability. Additionally, several discrepancies between the implementation of Qlapoti and the paper's pseudocode were never analyzed explicitly. We address these shortcomings and add further improvements, resulting in a new norm equation solving algorithm with negligible failure probability. Our C implementations shows 6x to 9x speedups compared to Qlapoti's norm equation solver, and 1.3x-2.1x speedups for a SQIsign NIST2 signature (depending on parameter sets).

Keywords: Isogenies, Quaternions, Ideal-to-Isogeny, SQIsign

1 Introduction

1.1 Context on ideal-to-isogeny translation

Isogeny-based cryptography is one of several approaches to modern post-quantum cryptography. While it is not yet practically used, signature schemes like SQIsign [1] and PRISM [5] offer the unique advantage of extremely small combined public key and signature sizes, and SQIsign is currently in the 3rd round of the call for additional post-quantum signatures by NIST [3]. This remains true even in light of the recent improvements to the best known generic attacks against isogeny-based cryptography introduced in [28].

Both of these signatures are based on the Deuring correspondence [14], which associates ideals in a quaternion algebra to isogenies between supersingular elliptic curves over a finite field. The secret quaternion ideals enable efficient computation of the signature isogeny, while all public material consists in elliptic

November 2026
Authors listed alphabetically, ams.org/profession/leaders/CultureStatement04.pdf.

curves and isogenies. In both SQIsign and PRISM, translating (secret) ideals to (public) isogenies and image curves is therefore a crucial building block of signing and key generation.

Ideal-to-isogeny subroutines have been the object of many improvements and optimizations in the past years. The first instantiations of SQIsign [11,12] relied on a generalization of the KLPT algorithm [19] to translate quaternion ideals into smooth-norm quaternion ideals which could then be translated and computed as one-dimensional smooth-degree isogenies. However, these isogenies had very large degree and the computation was thus quite slow, with signing times in the order of hundreds of milliseconds for the first NIST submission of SQIsign [8]. The development of the use of isogenies of 2- or 4-dimensional abelian varieties, for embedding one-dimensional isogenies allowed to improve the performance of the ideal-to-isogeny step, including in SQIsign, at the cost of more complex mathematical objects. Inspired by ideas introduced in Clapoti [26] for quadratic ideals, an efficient ideal-to-isogeny algorithm using two-dimensional isogenies was introduced in [6], resulting in a SQIsign variant based on 2-dimensional isogenies [6] (other variants exist, such as [25,15] based on slightly different ideas). These new quaternionic ideal-to-isogeny techniques later led to PRISM [5], which cannot be instantiated using only smooth-degree one-dimensional isogenies, as the required isogenies have large prime degree. The 2nd round NIST submission of SQIsign [1] adapted [6] and the quaternionic version of Clapoti, but in order to decrease the failure probability of the algorithm below 2^{-60}, important complexifications were required. These modifications increased considerably the memory requirement and the code complexity.

In comparison, Qlapoti [7] simplifies the translation of quaternionic ideals to isogenies considerably while reaching a better failure probability. In short, to translate an ideal I of a special starting order $\mathcal{O}_0$ into an isogeny embedded in a 2-dimensional 2^e-isogeny[1], it is necessary to find two ideals I_1, I_2 both equivalent to I and whose norms are such that

$$n(I_1) + n(I_2) = 2^e. \tag{1}$$

This step is referred to as the norm equation step. Then the endomorphism θ described by the product $I_2\bar{I}_1$ is evaluated at the special starting curve E_0 and the image determines the kernel of the 2-dimensional 2^e-isogeny through which the isogeny associated to I can be evaluated. We refer to this part as the isogeny-side of the ideal-to-isogeny translation. The main improvement of Qlapoti over the previous approach using Clapoti adaptations is that it gives an algorithm to solve exactly Eq. (1), while the Clapoti adaptations required to add additional scalar factors which made the isogeny side of the ideal-to-isogeny translation more complex and costly. Additionally, the norm equation solving algorithm in Qlapoti is fairly simple, compared to previous approaches.

[1] The choice of a power of 2 is made for efficiently, as 2-isogenies chain are the fastest to compute.

1.2 Shortcomings of Qlapoti and our improvements

Despite its attractiveness, the Qlapoti algorithm as presented in [7], exhibits serious drawbacks. We give a list of shortcomings that motivated our work, which addresses these issues.

1. The norm equation algorithm as described in the paper does not enforce that the solution translates to a computable 2^e-isogeny, since, if $n(I_1)$ and $n(I_2)$ are even, no kernel basis can be computed. Additionally, it does not guarantee that the first $(2,2)$-isogeny step of the two-dimensional isogeny to be computed is a gluing isogeny, i.e. an isogeny from a product of elliptic curves to an abelian variety which is not such a product. Handling non-immediate gluing requires to work with xy-arithmetic, thus leading to more complex isogeny code and variable-length $2^{e'}$ length 2-dimensional isogenies after the non-gluing steps, which are more difficult to optimize. Finally, as the norm equation side costs little compared to the isogeny side, it makes sense to handle additional complications on this faster side by rejecting unsuitable norm equation solutions. While the implementation section of Qlapoti mentions these issues, and their C implementation of the norm equation solver addresses it (which is necessary for its integration in the SQIsign NIST implementation's optimized code), the impact of these modification was not studied in the paper.

→ *We fill in this gap by providing a norm equation solver ensuring computable isogenies and immediate gluing at a minimal cost.*

2. The pseudocode presented in [7] is not optimized and not faithful to what is implemented. As a consequence, the impact of several of the optimizations used in the implementation are not properly analyzed. Furthermore, even with these optimizations, the norm equation step remains quite inefficient for some parameter sets. In particular for the SQIsign parameter set called Level 3, the norm equation solver from [7] takes nearly half of the runtime of the signing procedure.

→ *We provide an optimized implementation of our algorithm which outperforms the previous implementation by a factor 5 to 9. Contrarily to Qlapoti, the pseudocode we provide and analyze matches closely what we implemented, up to tiny and clearly stated changes which can only improve its success probability.*

3. The analysis of the failure probability of Qlapoti makes some assumptions which are too optimistic. In particular, the failure probability estimate given in [7, Table 3], is based on the heuristic that the minima of a two-dimensional lattice appearing in some part of the algorithm is following the Gaussian heuristic, which we show is false. This oversight cascade into some failure probability estimates that do not reflect the reality of the algorithm.

→ *We provide a thorough analysis of our new algorithm, including all its optimizations, and prove under plausible and experimentally verified heuristics that the failure probability is negligible for parameters relevant for cryptography, as shown in Table 1. Following this analysis, we believe the numbers presented in [7, Table 3] are off by a factor of at least* 2^{64}.

1.3 Outline of the paper

In order to achieve the improvements listed above, we modified almost every step in the norm equation solver of Qlapoti, resulting in a new algorithm we call *Qlapoty*. [2]

In Section 2, we define most notations and provide the necessary background on quaternions to understand the norm equation algorithm, of which we provide a detailed overview in Section 2.2. At a high level, we explain how the norm equation solver can be divided into two consecutive steps or loops (as they both iterate over a set of candidates until some suitable element is found). The input of the second loop is essentially the output of the first loop. We will treat and analyze these two loops somewhat independently.

We start with our variant of the second step of the algorithm in Section 3. In particular, this is where we enforce the geometric conditions efficiently. We identify a precise set of conditions on its input for this loop to succeed with high probability. Then, we explain in Section 4 when and how the input conditions of the second step can be met.

Finally, we put our two building blocks together to obtain our algorithm Qlapoty and analyze its global failure probability in Section 5. Benchmarks of our C implementation are given in Section 6.

Related work. While this work was finalized, we became aware of an independent and concurrent work [20] on improving Qlapoti. Interestingly, the proposed solutions are quite unrelated to ours. Indeed, most of that work focuses on exploiting all optimization possibilities provided by the extended Cornacchia algorithm. In contrast, our works minimise the use of the Cornacchia algorithm, so that we could use a Cornacchia algorithm working only on prime inputs without issues. For that reason, our improvements are more or less incompatible with the ones proposed in [20]. Furthermore, [20] does not address the non-immediate gluing problem and therefore works with variable length 2-dimensional isogenies, which is undesirable as explained above. Finally, it does not improve on the failure analysis of [7] and therefore suffers from the same issues.

Acknowledgements. The authors would like to thank the SQIsign NIST submission team, with special thanks to Pierrick Dartois, Luca De Feo, Jonathan Komada Eriksen, Riccardo Invernizzi, Marzio Mula & Lorenz Panny for the useful discussions.

Sina Schaeffler is supported by SNSF Consolidator Grant CryptonIs 213766.

[2] This is a shorthand for *Qlapoti v2* where the *i* and *v* fused into a *y*, which additionally looks like a roman *II*. The authors emphasize that Qlapoty is strictly pronounced /kla.pɔ.ti/, and is therefore phonetically identical to Clapotis [26], Clapoti [26], Klapoti [27], and Qlapoti [7], since that seems the word designating ideal-to-isogeny algorithms using embeddings in higher-dimensional isogenies.

2 Background and notations

We start with introducing some minimal mathematical background, followed by a detailed description of the Qlapoti algorithm as provided in [7]. In both of these parts we introduce notations used in the rest of the paper, for mathematical operators, but also for variables and parts of the algorithms.

Most notations from this section are reused lateron, often without recall.

2.1 Notations and mathematical background

Basic notations, and background on quaternions. We denote p a prime 7 (mod 8) and work over a finite field $\mathbb{F}_{p^2}$ on the geometric side, and in a quaternion algebra $\mathcal{B}_{p,\infty}$ ramified at p and infinity on the algebraic side. Let $\mathbf{i}, \mathbf{j}$ such that $(1, \mathbf{i}, \mathbf{j}, \mathbf{ij})$ is a basis of $\mathcal{B}_{p,\infty}$ with $\mathbf{i}^2 = -1$, $\mathbf{j}^2 = -p$ and $\mathbf{ij} = -\mathbf{ji}$. We denote $n(x)$ the (reduced) norm of a quaternion x, $\overline{x}$ its conjugate and $\mathrm{tr}(x)$ its (reduced) trace.

We understand ideals to be primitive integral left ideals of the maximal order $\mathcal{O}_0$ of basis $(1, \mathbf{i}, \frac{\mathbf{i}+\mathbf{j}}{2}, \frac{1+\mathbf{ij}}{2})$ of $\mathcal{B}_{p,\infty}$. The norm $n(I)$ of an ideal I is the greatest common divisor (gcd) of the norms of its elements, an its conjugate $\overline{I}$ is the set of conjugates of its elements. Two ideals I, J are said equivalent if there is $\zeta \in \mathcal{B}_{p,\infty}$ non-zero such that $I = J\zeta$. Since both I and J are integral, $\zeta^{-1} = \bar{\zeta}/n(\zeta) \in J$, and then $n(I) = n(J)n(\zeta)$. This equivalence $\sim$ defines an equivalence relation, and therefore equivalence classes of ideals.

On the geometric side, we consider E_0 described by $y^2 = x^3 + x$ over $\mathbb{F}_{p^2}$ which is a supersingular elliptic curve, with $E_0(\mathbb{F}_{p^2}) \cong \mathbb{Z}_{p+1}^2$. It is equipped with an abelian group structure and we denote as $E_0[m]$ the m-torsion subgroup. By the Deuring correspondence [14], the quaternions in $\mathcal{O}_0$ (or equivalently its principal ideals) can be mapped to $\mathrm{End}(E_0)$, and ideals to isogenies of domain E_0. Equivalent ideals correspond to isogenies of domain E_0 with isomorphic codomain curves. Let $\theta \in \mathrm{End}(E_0)$. Through the Deuring Correspondence, it can be written as an $\mathcal{O}_0$-element, and thus as

$$\theta = x_\theta + y_\theta \mathbf{i} + z_\theta \frac{\mathbf{i}+\mathbf{j}}{2} + w_\theta \frac{1+\mathbf{ij}}{2} = a_\theta + b_\theta \mathbf{i} + c_\theta \mathbf{j} + d_\theta \mathbf{ij}$$

where the first is expressed in canonical basis for $\mathcal{O}_0$ and the second in the classical Quaternion basis (with coefficient in $\frac{1}{2}\mathbb{Z}$).

Canonical isomorphism between $\mathbb{Z}^2$ and $\mathbb{Z}[\mathbf{i}]$. Throughout this paper, we use the canonical isomorphism $\Psi : \mathbb{Z}^2 \to \mathbb{Z}[\mathbf{i}]$ defined by $\Psi(s,t) = s + \mathbf{i}t$. Since $\mathbb{Z}[\mathbf{i}]$ is an Euclidean domain, we denote its efficiently computable greatest common divisor by $\gcd_{\mathbb{Z}[\mathbf{i}]}$. Additionally, we define the norm of $x + \mathbf{i}y$ as $\sqrt{x^2 + y^2}$.

2.2 Qlapoti

For a full exposition of Qlapoti and its norm equation, we refer to the paper [7]. Below we fix some notations for the rest of this article while providing

an overview of the algorithm. A precise description of the algorithm, following closely [7], is given in Algorithm 5.

Goal of the algorithm. Qlapoti has two main parameters: the prime characteristic p, and an exponent e. On input a left $\mathcal{O}_0$-ideal J in $\mathcal{B}_{p,\infty}$, the goal is to find $I_1 \sim J$ and $I_2 \sim J$ such that $n(I_1)+n(I_2) = 2^e$. This is equivalent to solving the norm equation $n(\beta_1) + n(\beta_2) = 2^e n(I)$ with $\beta_1, \beta_2 \in I$ for some $I \sim J$. The Qlapoti algorithm can find solutions to this equation efficiently when $p/2^e$ is not too big (at most logarithmic in p).

This allows to embed the isogenies corresponding to the ideals I_1, I_2 into a 2-dimensional 2^e-isogeny that can be efficiently computed from the endomorphism $\theta := \beta_2 \cdot \overline{\beta_1}/n(I) \in \mathcal{O}_0$. To help with that computation, 2^{e+2} torsion points are useful, so we consider henceforth that $p = 2^{e+2}f - 1$ for some small odd f.

Overview of the algorithm. The first step of the algorithm is to compute I, the smallest equivalent ideal to its input J. This ideal I is where all the computations will be made. The norm of I will always be denoted N.

The main idea of Qlapoti is to find β_1, β_2 of the form

$$\beta_\kappa = \gamma_\kappa \cdot N + \alpha \text{ with } \gamma_\kappa := (a_{\gamma_\kappa} + b_{\gamma_\kappa}\mathbf{i}) \in \mathbb{Z}[\mathbf{i}],$$

where α is a (second) generator of I such that $I = \mathcal{O}_0\langle \alpha, N\rangle$. As the computation of γ_1, γ_2 depends on α, at a high level, Qlapoti can be seen as a two-step algorithm:

1. Sample a good generator α.
2. Try to solve for γ_1, γ_2. If it fails, go back to the first step.

Throughout this work, these two steps will sometimes be referred to as the *first and second loops* of Qlapoti.

Description of the second loop and its requirements on α. Let us write $\alpha = \alpha_1 + \alpha_2\mathbf{j}$, with $\alpha_1, \alpha_2 \in \mathbb{Z}[\mathbf{i}]/2$, and $r = n(\alpha)/N$. We furthermore write $\alpha_1 = a_\alpha + b_\alpha\mathbf{i}$ where a_α and b_α are rationals.

For loop 2, it was shown in [7] that if β_1, β_2 is a solution to the norm equation given by γ_1, γ_2 for α fixed, then there are $s, t, s_c, t_c \in \mathbb{Z}$ such that $\Psi(s,t) = \gamma_1+\gamma_2$, and $\Psi(s_c, t_c) = \gamma_1 - \gamma_2$ with (s,t) a short solution to the equation $2a_\alpha s + 2b_\alpha t = 2^e - 2r \pmod N$, and $s_c^2 + t_c^2 = z$ with $z = 2(2^e - 2r - 2a_\alpha s - 2b_\alpha t)/N - s^2 - t^2$. Once α is fixed, the equation is thus solved by computing such s, t, s_c, t_c and then reconstructing γ_1, γ_2 as $\gamma_1 = a_{\gamma_1} + \mathbf{i}b_{\gamma_1}$ and $\gamma_2 = a_{\gamma_2} + \mathbf{i}b_{\gamma_2}$ using $a_{\gamma_1} = (s_c+s)/2$, $b_{\gamma_1} = (t_c + t)/2$, $a_{\gamma_2} = s - a_{\gamma_1}$, $b_{\gamma_2} = t - b_{\gamma_1}$. For the values $a_{\gamma_1}, a_{\gamma_2}, b_{\gamma_1}, b_{\gamma_2}$ to be integers, we additionally need $s_c = s \pmod 2$ and $t_c = t \pmod 2$.

For this resolution method to work, the norm r and the integers (s,t) need to be short so that $z > 0$. Then s_c, t_c can be found with integer sum-of-two-squares solver, such as Cornacchia's algorithm (see [24] for a modern description).

Solutions will exist if s, t and z satisfy a few conditions modulo 4 (see [7, Algorithm 2] for details). Finding short (s, t) amounts to solving a close vector problem (CVP)[3] in a rank 2 lattice $\Lambda(\alpha)$ associated to α. This lattice is defined as $\Psi^{-1}\left(\ker \operatorname{tr}((s - \mathbf{i}t)\alpha)\right) \mod N$, or equivalently, as the set of solutions to the equation $2a_\alpha s + 2b_\alpha t = 0 \mod N$. For the CVP resolution to succeed in finding a small enough solution, one needs the lattice $\Lambda(\alpha)$ to be balanced (i.e. to not contain a very small vector). This is the main requirement on α and the main cost of the first loop is the reduction of this 2D lattices for each candidate α.

Beyond the norm equation: rejections for geometric requirements. It is in theory always possible to compute an isogeny from a pair of ideals I_1, I_2 solution of the norm equation. However, the practical computation of the two-dimensional isogeny of degree $(2^e, 2^e)$ from the associated endomorphism θ is more complicated in some cases (e.g. when θ has even norm). These cases must thus be avoided, as explained in the introduction.

As a result, some solutions are rejected during the norm equation solving process. This is the case for instance in the original implementation of Qlapoti in SQIsign's NIST submission's C code [7, Section 5]. In particular, diagonal isogenies (that means, isogenies between products of elliptic curves) at the start of the two-dimensional 2^e-isogeny chain are creating complications and should be avoided. One of our contributions is to express these geometric requirements in terms of the quaternionic representation of θ and to show how to catch these cases early in the algorithm.

Minimizing the number of calls to the first loop. A central point in the implementation of Qlapoti (but never analyzed properly in [7]) is the reuse of the same lattice for different α. More specifically, given an initial candidate α_0, one can consider its scalar multiples $\alpha = \lambda\alpha_0$ where λ is an integer coprime to N as suitable candidates. We then have that $\Lambda(\alpha) = \Lambda(\alpha_0)$, but $n(\alpha) = \lambda^2 n(\alpha_0)$. This yields different target vectors for the CVP involved in the search for short s, t, thus acting as a randomization mechanism without having to run the first loop again. Note that there is a bound on λ, that we denote B_λ as $z = (2^e - \lambda^2 n(\alpha_0)/N - 2\lambda a_{\alpha_0} s - 2\lambda b_{\alpha_0} t)/N - s^2 - t^2$ needs to remain positive.

This dichotomy justifies why the two-step overview of Qlapoti we gave earlier in this section is particularly relevant. We can adjust it with the above idea of considering several α stemming from the same α_0. The two loops description of the algorithm becomes:

1. Find an α_0 for which the lattice $\Lambda(\alpha_0)$ is balanced.
2. Iterate λ coprime to N and run the second loop on $\alpha = \lambda\alpha_0$ to attempt finding a short (s, t) such that z can be decomposed in a sum of two squares $s_c^2 + t_c^2$ efficiently by Cornacchia's algorithm. If no solution is found, go back to the first step.

[3] Note that we don't need to find a closest vector, only one that is close enough.

This description omits several subtle additional conditions on α_0, and we will provide a very concrete definition of what is meant by "balanced" later on.

Remark 1. Another method to resample α from an initial α_0 would be to have $\alpha \in \alpha_0 + N\mathbb{Z}[i]$. Indeed, at first glance, we have that $\Lambda(\alpha) = \Lambda(\alpha_0)$ while $2^e - 2n(\alpha) \neq 2^e - 2n(\alpha_0) \mod N$ and thus vary the (s,t) solutions just as when $\alpha \in \alpha_0\mathbb{Z}$. Nevertheless, it occurs that this method does not work as this sampling method is very likely to not randomize the value of z. We refer to Appendix B for a proof of this result.

3 Loop 2: looking for suitable (s,t) given a fixed α_0

The goal of this section is to explain how one can look for a suitable pair of elements (s,t) more efficiently. This is the main part of Qlapoty's second loop. Mainly, we identify bad conditions as early as possible in the algorithm to avoid useless computations. In the process, we also identify necessary conditions on the generator α_0 that will prove crucial in our revision of the first loop in Section 4.

In Section 3.1, we first look at the geometric conditions on the norm equation solution and how they affect the computation. Then, in Section 3.2, we turn to the CVP computation that lies at the heart of the computation of (s,t). We combine these pieces in Section 3.3 to a new Loop2 algorithm analyze its failure probability.

3.1 Embedding the geometric conditions in the search for (s,t)

As explained in the previous section, the Qlapoti norm equation is part of a broader ideal-to-isogeny mechanism which requires to compute a two-dimensional isogeny from the endomorphism $\theta \in \mathrm{End}(E_0)$. Practical considerations of this isogeny computation impose *geometric conditions* on the output of our norm equation.

Geometric conditions on the endomorphism θ. Let (P,Q) be a basis of $E_0[2^{e+2}]$ and let $\theta \in \mathrm{End}(E_0)$ be a solution of the Qlapoti norm equation, we then have that $\deg(\theta) = u(2^e - u)$ for some integer u. By Kani's Lemma [18], we can construct, using θ, $\mathcal{B}$ a basis of $\Phi : E_0^2 \to E \times E'$ a $(2^e, 2^e)$-isogeny. $\mathcal{B}$ is of the following form.

$$\mathcal{B} := \left\{ \left([-u]P, \theta(P)\right), \left([-u]Q, \theta(Q)\right) \right\}$$

For this construction to be valid and easy to compute, we need 2 conditions:

1. For $\mathcal{B}$ to be well defined, we need $\gcd(u, 2^e - u) = 1$, i.e. u is odd.
2. We desire the first $(2,2)$-isogeny in the chain computing Φ to be a gluing isogeny, i.e an isogeny $\psi : E_0^2 \to J_1$, where J_1 is the jacobian of an hyperelliptic curve of genus 2.

By Deuring correspondence [14], we can translate the first condition to the quaternion side as sampling β_1 such that $n(\beta_1)/n(I)$ is odd. The second condition can be expressed as follows.

Lemma 1. *Let $\theta \in End(E_0)$ be an endomorphism of norm $u(2^e - u)$ for u odd. Let Φ be the $(2^e, 2^e)$ isogeny corresponding to the basis $\mathcal{B}$.*

ψ the first isogeny of the (2,2)-chain of Φ is a gluing $\iff \theta \in \mathcal{O}_0 \backslash \mathbb{Z}[\mathbf{i}, \mathbf{j}]$

Although the condition introduced in Lemma 1 was mentioned in the original paper [7, Section 5.2], a detailed explanation was lacking. We provide a detailed proof of this result in Appendix A.1. The rest of this section is dedicated to understanding how the two required conditions on θ can be enforced efficiently during the computation of (s, t).

Detecting gluing before Cornacchia. Our first goal is to ensure that the $\theta \in \mathrm{End}(E_0)$ stemming from our solution of the Qlapoti norm equation Eq. (1) is such that $\theta \notin \mathbb{Z}[\mathbf{i}, \mathbf{j}]$. We reuse the same notation as in Section 2.2 and refer to it for the relations between the different variables. We now express Lemma 1 as follows.

Lemma 2.

$$\theta \in \mathbb{Z}[\mathbf{i}, \mathbf{j}] \subset \mathcal{O}_0 \Longleftrightarrow (s + \mathbf{i}t) \cdot \overline{\alpha} \in \mathbb{Z}[\mathbf{i}, \mathbf{j}]$$

Proof.

$$\begin{aligned}
\theta &= (\gamma_2 \cdot N + \alpha) \cdot \overline{(\gamma_1 \cdot N + \alpha)}/N \\
&= \underbrace{N\gamma_2\overline{\gamma_1} + n(\alpha)/N}_{\in \mathbb{Z}[\mathbf{i}]} + \alpha\overline{\gamma_1} + \gamma_2\overline{\alpha} \\
&= \underbrace{N\gamma_2\overline{\gamma_1} + n(\alpha)/N}_{\in \mathbb{Z}[\mathbf{i}]} + \underbrace{\alpha \cdot \overline{\gamma_1} - \gamma_1 \cdot \overline{\alpha}}_{\in \mathbb{Z}[\mathbf{i},\mathbf{j}]} + \underbrace{(s + \mathbf{i}t) \cdot \overline{\alpha}}_{\in \frac{1}{2}\mathbb{Z}[\mathbf{i},\mathbf{j}]}
\end{aligned}$$

□

Note that some variables are striking by their absence, and those are γ_κ, meaning that the output of Cornacchia in the norm equation solver plays no role in this specific condition. Thus, we can determine whether the output will be acceptable prior to running Cornacchia and combine this with the conditions which ensure the existence of a solution of the Qlapoty norm equation. In the rest of this section, let us denote k for $(2^e - 2r - 2a_\alpha s - 2b_\alpha t)/N$.

Lemma 3. *Let $\alpha = a_\alpha + b_\alpha\mathbf{i} + c_\alpha\mathbf{j} + d_\alpha\mathbf{ij} = x_\alpha + y_\alpha\mathbf{i} + z_\alpha\frac{\mathbf{i}+\mathbf{j}}{2} + w_\alpha\frac{1+\mathbf{ij}}{2} \in \mathcal{O}_0$, with $n(\alpha) = Nr$. Let (s, t) be solution of the equation $2a_\alpha s + 2b_\alpha t = 2^e - 2r \mod N$, with $kN = 2^e - 2r - 2a_\alpha s - 2b_\alpha t$ and $z = 2k - s^2 - t^2$. Finally, let s_c, t_c be such that $s_c^2 + t_c^2 = z$.*

Then $(s + \mathbf{i}t) \cdot \overline{\alpha} \in \mathcal{O}_0 \backslash \mathbb{Z}[\mathbf{i}, \mathbf{j}]$ and $s_c = s \mod 2$ and $t_c = t \mod 2$ (up to permutation of s_c, t_c) [4] if and only if one of the following two conditions holds:

[4] This second condition is required to ensure the existence of a solution of the Qlapoti norm equation, as $a_{\gamma_1} = (s_c + s)/2$ and $b_{\gamma_1} = (t_c + t)/2$.

– $z_\alpha + w_\alpha$, $s+t$ *and* k *odd and* $z = 1 \mod 4$.
– N, z_α, w_α *and* k *are odd,* $s+t = 1 \mod 2$ *and* $z = 1 \mod 4$.

The proof is computational and detailed in Appendix A.2.

Detecting odd solutions before Cornacchia. Now that we know how to ensure an immediate gluing, we must still ensure that the two equivalent ideals output by the norm equation solver are of odd norm. This also can be done prior to the call to Cornacchia's algorithm. First, we note that most valid solutions detailed in the proof of Lemma 3 require that $z_\alpha + w_\alpha = 1 \mod 2$. The only exception consists in the case when N, z_α, w_α and k are odd and r is even. We now show this latter case is not exploitable as it will lead to even norm ideals. To demonstrate this in Corollary 1, we rely on the following result.

Proposition 1. *Let* $I_1 \sim I_2$ *be left* $\mathcal{O}_0$*-ideal such that* $n(I_1) + n(I_2) = 2^e$. *Let* $\mathcal{O}_0\theta = I_2\overline{I_1}$ *with* $\theta = a_\theta + b_\theta\mathbf{i} + c_\theta\mathbf{j} + d_\theta\mathbf{ij}$ *primitive in* $\mathcal{O}_0$. *Then,*

$$\gcd(n(I_1), n(I_2)) = 1 \Longleftrightarrow \exists!\, x \in \{2a_\theta, 2b_\theta, 2c_\theta, 2d_\theta\} \text{ s.t. } x = 2 \mod 4$$

The proof is computational and detailed in Appendix A.2.

Corollary 1. *Let* I_1, I_2 *be a solution of the Qlapoti norm equation Eq. (1) such that* N, z_α, w_α *and* k *are odd and* r *is even. Then* $\gcd(n(I_1), n(I_2)) \geq 2$.

Proof. Following the proof of Lemma 2, we obtain

$$2\theta = \underbrace{2r + 2N\gamma_2\overline{\gamma_1} + 2(\alpha \cdot \overline{\gamma_1} - \gamma_1 \cdot \overline{\alpha})}_{\in 2\mathbb{Z}[\mathbf{i},\mathbf{j}]} + \underbrace{2(s + \mathbf{i}t) \cdot \overline{\alpha}}_{\in \mathbb{Z}[\mathbf{i},\mathbf{j}]}$$

Thus, the odd coefficients of 2θ are the same as the odd coefficients of $2(s+\mathbf{i}t)\cdot\overline{\alpha}$. But, as z_α, w_α and $s+t$ are odd, we have that $2(s+\mathbf{i}t)\cdot\overline{\alpha} = 1+\mathbf{i}+\mathbf{j}+\mathbf{ij} \mod 2$ proving our lemma following Proposition 1. □

We now consider the case where $z_\alpha + w_\alpha = 1 \mod 2$ (as shown above this is the only case leading to exloitable solutions), and show that the parity of $n(\theta)$ can be determined without knowing the output of the Cornacchia algorithm.

Lemma 4. *Let* α, s, t, k, z *be a solution of Lemma 3 such that* $z_\alpha + w_\alpha = 1 \mod 2$. *Let* $s_c^2 + t_c^2 = z$ *and let* θ *be the corresponding solution of the Qlapoti norm equation. Then,* $n(\theta)$ *is odd if and only if:*

$$r + \frac{k-1}{2} = 1 \mod 2$$

Proof. To ensure $n(\theta)$ is odd, it suffices to ensure $n(\beta_\kappa)/N = 1 \mod 2$. Looking at β_1, this can be rewritten as:

$$N(a_1^2 + b_1^2) + r + 2a_\alpha a_1 + 2b_\alpha b_1 = 1 \mod 2 \tag{2}$$

where $a_1 = (s_c + s)/2$ and $b_1 = (t_c + t)/2$. We can choose $s_c = s \mod 2$ and $t_c = t \mod 2$ since:

$$s_c^2 + t_c^2 = z = 2k - s^2 - t^2$$

As $z = 1 \mod 4$, we must have $s_c + t_c = 1 \mod 2$. Since all odd integers have square equal to $1 \mod 8$, if we assume $s_c = 0 \mod 2$, then $s_c = 0 \mod 4$ if $z = 1 \mod 8$, while $s_c = 2 \mod 4$ if $z = 5 \mod 8$. i.e. $s_c = (z-1)/2 \mod 4$.

We also observe:

$$N(a_1^2 + b_1^2) = N\left((s_c + s)^2 + (t_c + t)^2\right)/4 = N(k + s_c s + t_c t)/2$$

Rewriting (2) as:

$$N\frac{k + s_c s + t_c t}{2} + r + 2a_\alpha \frac{s_c + s}{2} + 2b_\alpha \frac{t_c + t}{2} = 1 \mod 2 \tag{3}$$

and assuming $2b_\alpha = z_\alpha = 0 \mod 2$ (and thus $2a_\alpha = w_\alpha = 1 \mod 2$):

- **Case** $N = 0 \mod 2$**:** As shown in Lemma 3, $2a_\alpha s + 2b_\alpha t = 0 \mod 2$, implying $s = s_c = 0 \mod 2$. Using the parities of s, s_c and t, t_c with odd k, equation (3) simplifies to:

$$\begin{aligned} 1 &= r + \frac{s_c}{2} + \frac{s}{2} = r + \frac{z-1}{4} + \frac{s}{2} = r + \frac{2k - s^2 - t^2 - 1}{4} + \frac{s}{2} \\ &= r + \frac{2(k-1) - s^2}{4} + \frac{s}{2} = r + \frac{k-1}{2} - \frac{s^2}{4} + \frac{s}{2} = r + \frac{k-1}{2} \mod 2. \end{aligned}$$

- **Case** $N = 1 \mod 2$**:** Since $k = 1 \mod 2$, necessarily $2a_\alpha s + 2b_\alpha t = 1 \mod 2$, implying $s = s_c = 1 \mod 2$. Equation (3) becomes:

$$\begin{aligned} 1 &= N\frac{k + s_c s}{2} + r + 2a_\alpha \frac{s_c + s}{2} = \frac{k + s_c s + s_c + s}{2} + r \\ &= \frac{k-1}{2} + \frac{(s_c + 1)(s + 1)}{2} + r = \frac{k-1}{2} + r \mod 2. \end{aligned}$$

The case where $2a_\alpha = 0 \mod 2$ is analog. □

Thus, we can always determine if the endomorphism θ will have an odd norm prior to running Cornacchia. Combining Lemma 3 and Lemma 4 implies that we must enforce k is odd with $k \mod 4$ depending on $r \mod 2$. By slightly adapting the solution method of the equation modulo N, we can always enforce that $k \mod 4$ is correct without rejection sampling. Once this is achieved, the only remaining rejections are due to cases when $s = t \mod 2$, provided $z > 0$.

To do so, we remark that solving the modular equation $2^e - 2r - 2a_\alpha s - 2b_\alpha t = kN = (4k' + \delta_r)N$ with $\delta_r = 1$ if $r_0 = 1 \mod 2$ and $\delta_r = 3$ if $r_0 = 0 \mod 2$ is equivalent to solving the modular equation $2^e - 2r - 2a_\alpha s - 2b_\alpha t - \delta_r N = 0 \mod 4N$. This equation is a modular equation just as $2^e - 2r - 2a_\alpha s - 2b_\alpha t = kN$ and can therefore also be solved as a close vector problem in a lattice, yielding solutions (s, t) of comparable size (at most larger by a factor 4). Furthermore, as $\alpha = \lambda\alpha_0$, we have that $r = n(\alpha)/N = \lambda^2 r_0$ and thus its parity is fixed for all odd λ. This condition is therefore easy to enforce in practice.

3.2 On computing small (s,t) solutions

In this section, we turn to finding small (s,t) solutions of $2a_\alpha s + 2b_\alpha t = c \mod N$. In particular, with Lemma 5 below, we show that a solution can always be found when $\gcd(\mathrm{tr}(\alpha_0), \mathrm{tr}(\mathbf{i}\alpha_0), N) = 1$. (This condition is in fact minimal to ensure the existence of a solution (s,t)). To find this solution, we work with the lattice

$$\Lambda(\alpha_0) := \left\{ \Psi^{-1}\left(\ker\left(\sigma \mapsto \mathrm{tr}(\overline{\sigma}\alpha_0)\right) \mod 4N\right)\right\}$$

which is defined using Ψ the isometry between $\mathbb{Z}^2$ and $\mathbb{Z}[\mathbf{i}]$. $\Lambda(\alpha_0)$ is equal to

$$\left\{(s,t) \in \mathbb{Z}^2 \middle|\ 2a_\alpha s + 2b_\alpha t = 0 \mod 4N\right\}$$

Lemma 5. *Take $N \in \mathbb{N}$, $0 < a, b, d < 4N$ with $\gcd(a, b, 4N) = 1$ and let $\Lambda \subset \mathbb{Z}^2$ be the lattice of elements (s,t) satisfying $as + bt = 0 \mod 4N$. Let $g_a = \gcd(a, 4N)$, $g_b = \gcd(b, 4N)$ and $a = g_a \cdot a', b = g_b \cdot b'$. Then,*

$$\Lambda = \begin{pmatrix} c_a \\ g_a \end{pmatrix}\mathbb{Z} + \begin{pmatrix} 4N/g_a \\ 0 \end{pmatrix}\mathbb{Z} = \begin{pmatrix} g_b \\ c_b \end{pmatrix}\mathbb{Z} + \begin{pmatrix} 0 \\ 4N/g_b \end{pmatrix}\mathbb{Z}$$

with $c_a = -b/a' \mod 4N/g_a$ and $c_b = -a/b' \mod 4N/g_b$. Furthermore, let $g = \gcd(a,b)$, u, v such that $au + bv = g$ and let w, l such that $wk + 4Nl = 1$. Then,

$$as + bt = d \mod 4N \iff \begin{pmatrix} s \\ t \end{pmatrix} \in \Lambda + \begin{pmatrix} dwu \\ dwv \end{pmatrix}.$$

Proof. We see that the 2 vectors of Λ are linear independent and solutions of the modular equation, as $a \cdot 4N/g_\alpha = a' \cdot 4N = 0 \mod 4N$ and $ac + bg_\alpha = -bg_\alpha + bg_\alpha = 0 \mod 4N$. Furthermore, Λ is of volume $4N$. Hence, Λ is the lattice of the solution (s,t) such that $as + bt = 0 \mod 4N$.

By definition, we have that $awu + bwv = w(au + bv) = wg = 1 \mod 4N$. Therefore, for all d, we have that dwu, dwv is solution of $as + bt = d \mod 4N$ and we can write its affine lattice of solution as $\Lambda + \binom{dwu}{dwv}$. □

Using Minkowski reduced basis of the lattices of Lemma 5, we can use the Babai's rounding algorithm [4] to efficiently compute (s,t) solution of the equation $as + bt = d \mod 4N$ of small norm, i.e.

$$s^2 + t^2 \leq \frac{\left(\lambda_1\left(\Lambda(\alpha_0)\right) + \lambda_2\left(\Lambda(\alpha_0)\right)\right)^2}{4} \tag{4}$$

This yields the ComputeSmallSolutions algorithm that we explicit in Appendix D. By Eq. (4), for the norm of the found (s,t) to be small enough, it is important that the lattice $\Lambda(\alpha_0)$ is balanced.

3.3 Algorithm & Analysis

Combining all results of this section, we describe the second loop of Qlapoty Algorithm 1. This algorithm takes as input a second generator $\alpha_0 \in \mathcal{O}_0 \backslash \mathbb{Z}[\mathbf{i}, \mathbf{j}]$ and such that $\mathrm{tr}(\alpha_0)$ and $\mathrm{tr}(\mathbf{i}\alpha_0)$ have different parity, together with its lattice $\Lambda(\alpha_0) := \left\{\Psi^{-1}\left(\ker\left(\sigma \mapsto \mathrm{tr}(\overline{\sigma}\alpha_0)\right) \mod 4N\right)\right\}$, and outputs a solution of the Qlapoti norm equation Eq. (1). Otherwise, it returns $\perp$.

This section furthermore assumes that α_0 is such that the second minimum of $\Lambda(\alpha_0)$ is smaller[5] than $\sqrt{512N/\pi}$, and that $\gcd(\mathrm{tr}(\alpha_0), \mathrm{tr}(\mathbf{i}\alpha_0), N) = 1$. We are going to rely heavily on this balancedness of $\Lambda(\alpha_0)$ to justify that the solution (s, t) is small enough. The other condition comes from Lemma 5 to ensure that the resolution is always possible.

Algorithm 1 Loop2

Require: $\alpha_0 \in \mathcal{O}_0 \backslash \mathbb{Z}[\mathbf{i}, \mathbf{j}]$ a generator of ideal I, smallest of its class of norm N, such that $\mathrm{tr}(\alpha_0), \mathrm{tr}(\mathbf{i}\alpha_0)$ have different parity such that $\gcd(\mathrm{tr}(\alpha_0), \mathrm{tr}(\mathbf{i}\alpha_0), 4N) = 1$, together with its corresponding reduced lattice $\Lambda(\alpha_0)$.

Ensure: A solution β_1, β_2, θ of the norm equation satisfying the geometric conditions of Section 3.1.

$r_0 \leftarrow n(\alpha_0)/N$
$\delta_r \leftarrow 1$ **if** $r_0 = 1 \mod 2$, $\delta_r \leftarrow 3$ otherwise.
$\lambda \leftarrow -1$ ▷ test odd solution only
$B_\lambda \simeq \sqrt{2^{e-2}/r_0}$.
while $\lambda < B_\lambda$ **do**
  $\lambda \leftarrow \lambda + 2$
  if $\gcd(2N, \lambda) \neq 1$: **continue**
  $d \leftarrow 2^e - 2\lambda^2 r_0 - \delta_r N$
  $(s, t) \leftarrow$ ComputeSmallSolutions$(\Lambda(\alpha_0), \lambda, d, 4N)$ ▷ (s, t) such that $2a_{\alpha_0}\lambda s + 2b_{\alpha_0}\lambda t = 2^e - 2\lambda^2 r_0 - \delta_r N \mod 4N$
  if $s = t \mod 2$: **continue** ▷ Ensure $z = 1 \mod 4$.
  $k \leftarrow (2^e - 2\lambda^2 r_0 - 2\lambda a_{\alpha_0} s - 2\lambda b_{\alpha_0} t)/N$
  $z \leftarrow 2k - s^2 - t^2$
  if $z < 0$: **continue**
  sol $\leftarrow$ Cornacchia(z)
  if sol $= \perp$: **continue**
  $(s_c, t_c) \leftarrow$ sol
  if $s_c \neq s \mod 2$ **then** swap s_c, t_c
  $a_1 \leftarrow (s_c + s)/2, \quad b_1 \leftarrow (t_c + t)/2$
  $a_2 \leftarrow s - a_1, \quad b_2 \leftarrow t - b_1$
  $\beta_1 \leftarrow N(a_1 + \mathbf{i}b_1) + \lambda\alpha_0, \quad \beta_2 \leftarrow N(a_2 + \mathbf{i}b_2) + \lambda\alpha_0$
  $\theta \leftarrow (\beta_2 \cdot \overline{\beta_1})/N$
  return β_1, β_2, θ
end while
return $\perp$

[5] or equivalently, that $\lambda_1(\Lambda(\alpha_0)) \geq \sqrt{N/2}$ by the second Minkowski theorem.

We shall now study the success probability of Algorithm 1. For it to succeed, we need at least one odd $\lambda \leq B_\lambda$ to succeed (i.e., finish the loop). In this subsection, we will characterize ρ_λ, the probability that a single λ succeeds.

First conditions. The first two conditions under which an odd λ can fail are easy to characterize. First, it must satisfy $\gcd(\lambda, 2N) = 1$, which occurs with probability $8/\pi^2 + O(\log(N)/N) \simeq 8/\pi^2$ [16, §18.5]. Second, s and t must have different parities, which occurs with probability $1/2$. We have experimentally verified both of these heuristic values using our sage implementation.

Positive z. We then focus on the condition $z > 0$, which is equivalent to

$$2^{e+1} \geq \underbrace{4\lambda^2 r_0}_{(1)} + \underbrace{4\lambda a_{\alpha_0} s + 4\lambda b_{\alpha_0} t}_{(2)} + \underbrace{N(s^2+t^2)}_{(3)}, \tag{5}$$

and can be split into three parts:

(1) This condition only depends on α_0 and can thus be easily bounded by the choice of B_λ, as $B_\lambda \leq \sqrt{2^{e-2}/r_0}$ ensures that $(1) \leq 2^e$.
(2) This condition can also be easily upper-bounded. Indeed, following Babai's rounding algorithm on a Minkowski-reduced basis (see Eq. (4)), we have that

$$s^2 + t^2 \leq \frac{(\sqrt{16N/\pi} + \sqrt{512N/\pi})^2}{4} = \frac{(2^2 + 2^{4.5})^2 N}{4\pi} \leq 57N \tag{6}$$

using the first Minkowski theorem and the second minimum of $\Lambda(\alpha_0)$. We therefore have that both s and t can be upper-bounded by $\sqrt{57N}$. Similarly, we can enforce (as demonstrated in Proposition 2) that $2a_{\alpha_0}, 2b_{\alpha_0} \leq 2N$. Combining everything, we get that $(2) \leq 8\sqrt{57}N^{3/2}\lambda \ll 2^{e+1}$ for $\lambda \ll \sqrt{N}$, which is always true in our case[6]. Hence, its contribution to Eq. (5) is negligible.
(3) The contribution of (3) is the most impactful. Indeed, by Eq. (6), we have that $(3) \leq 57N^2 \leq 57 \cdot \frac{8}{\pi^2} p \leq \frac{456}{\pi^2} p$, which is greater than 2^e. Consequently, to ensure that $z > 0$, we need to have $s^2 + t^2$ small.

To better characterise such events, we reuse the analytical framework established in [7, Section 4]. More specifically, we use two probability distribution functions whose probability density functions are denoted by f and cumulative distribution functions by F:

– f_δ with $\delta = N/\sqrt{p}$, representing the random variable over the domain of all norms of all left $\mathcal{O}_0$-ideals which are smallest in their class.

[6] More specifically, in the case where $B_\lambda \simeq \sqrt{N}$, the algorithm would have succeeded long before this becomes significant as we see in the rest of this section and in Section 5.2.

– For fixed N, a, b such that $\gcd(a, b, 4N) = 1$, f_ϵ with $\epsilon = \sqrt{s^2 + t^2/N}$, representing the random variable over the domain $c \in \mathbb{Z}_N$ odd, where (s, t) is a solution of small norm to the linear equation $as + bt = c \mod 4N$ computed using Babai's algorithm and such that $\lambda_1(\Lambda) \geq \sqrt{N/2}$.

Remark 2. Although the first function f_δ is identical to that in [7, Section 4], the second function f_ϵ is not. We believe this function to be more representative of the reality of the second loop of Qlapoty. Crucially, the condition $\lambda_1(\Lambda) \geq \sqrt{N/2}$ ensures that the rank-2 lattice $\Lambda(\alpha_0)$ follows the Gaussian heuristic. We refer to its analysis in Appendix A.5.

Remark 3. Our assumption that the target c in Loop2 behaves like a random function comes from an empirical analysis of distribution of our sampling function $\lambda \mapsto \lambda^{-1}(2^e \pm N) - 2\lambda r_0 \mod 4N$, analysis available in our sage code.

Setting (2) to be negligible and $(1) \leq 2^e$, we can express Eq. (5) as $2^e \geq N(s^2 + t^2) = \frac{N^2(s^2+t^2)p}{Np} = \delta^2\epsilon^2 p \simeq \delta^2\epsilon^2 2^{e+2} f$ as we have $p = 2^{e+2}f - 1$. Thus, we have that

$$4f \leq \delta^{-2}\epsilon^{-2} \Longleftrightarrow \epsilon \leq (2\delta\sqrt{f})^{-1}.$$

Therefore, to obtain $z > 0$, we must get an (s, t) such that $\epsilon \leq (2\delta\sqrt{f})^{-1}$, which occurs with probability $F_\epsilon((2\delta\sqrt{f})^{-1})$. In all the cases we are interested in, as $\lambda_1(N) \geq \sqrt{N/2}$, we have that $2\delta\sqrt{f} \geq \sqrt{8}$, which, based on our experimental study of f_ϵ in Appendix A.5, ensures that we can restrict ourselves in the case where $f_\epsilon(x)$ is linear. Hence, we have that

$$F_\epsilon\big((2\delta\sqrt{f})^{-1}\big) \simeq \frac{\pi}{4}(2\delta\sqrt{f})^{-2} = \frac{\pi}{16\delta^2 f}.$$

Cornacchia. Finally, assuming $z > 0$, z must have an output by Cornacchia's algorithm. As using Cornacchia for decomposing z into a sum of two squares requires the computation of a square root of minus 1 modulo z, a factorization of z must be known. For our analysis, we thus assume that Cornacchia's algorithm only returns a solution when z is prime (and $z = 1 \mod 4$, which is ensured by Lemma 3). This is a worst-case assumption, attempts at factorising non-prime z can only improve the sucess probability. Since $z = 2k - s^2 - t^2 \approx \sqrt{p}/(2f\delta)$, the probability that z is prime is approximately $\big(\log(\sqrt{p}/(2f\delta))\big)^{-1}$.

Putting everything together, we obtain that ρ_λ, the success probability for a given λ, can be bounded as

$$\rho_\lambda \geq \frac{8}{\pi^2} \cdot \frac{1}{2} \cdot \frac{\pi}{16\delta^2 f} \cdot \frac{1}{\log(\sqrt{p}/(2f\delta))} \approx \frac{1}{4\pi\delta^2 f \log(\sqrt{p}/(2f\delta))}. \tag{7}$$

Equation (7) will then be used in Section 5.2 to upperbound the failure probability of Qlapoty.

Remark 4. Note that our analysis of ρ_λ and consequently our approximation Eq. (7) is conservative. It therefore serves as a strict lower bound that does not fully reflect the practical performance of the implementation.

4 Loop 1: sampling suitable generators α_0

This section explains how to perform the first loop of Qlapoti efficiently. We recall that the goal of the first loop is to find an element $\alpha_0 \in I$ such that:

1. $\alpha_0 \notin \mathbb{Z}[\mathbf{i}, \mathbf{j}]$ such that $\mathrm{tr}(\alpha_0)$ and $\mathrm{tr}(\mathbf{i}\alpha_0)$ have different parity.
2. $r_0 = n(\alpha_0)/N$ is as small as possible.
3. The lattice $\Lambda(\alpha_0)$ is balanced (i.e. $\lambda_2(\Lambda(\alpha_0)) \leq B_2(N)$ for some $B_2(N)$).
4. $\gcd(\mathrm{tr}(\alpha_0), \mathrm{tr}(\mathbf{i}\alpha_0), N) = 1$

The first condition is derived from Section 3.1. The impact of the second and third conditions on the success probability of Qlapoti's second loop was analyzed thoroughly in Section 3.3. The value of r_0 is directly tied to the number of possible candidates through B_λ, and the bound $B_2(N)$ on the second minimum of $\Lambda(\alpha_0)$ was crucial in estimating F_ε. The estimates from Section 3.3 assume that $B_2(N) = \sqrt{512N/\pi}$, but we use the generic $B_2(N)$ for most of this section. The fourth condition comes from Lemma 5.

It is to be noted that in [7], the specific procedure to sample α or α_0 is not well specified, but we will see that it is in fact of great importance.

First, in Section 4.1, we introduce a simple and minimal approach to sample small elements $\alpha_0 \in I$. In particular, this essentially deals with the first two conditions. Then, in Section 4.2, we analyze the lattice $\Lambda(\alpha_0)$ and sketch a strategy to find suitable α_0 satisfying the balancedness condition. We prove that such an α_0 can almost always be computed in Section 4.3, and provide a concrete algorithm to find it. The fourth and final condition on α_0 follows naturally from that algorithm.

4.1 Finding small generators $\alpha_0 \in I$

Using the HNF basis. To sample small elements in I, it seems natural to use a reduced basis. We first remark that in the specific case where I is the smallest ideal in its equivalence class, a very short basis is directly obtained with the HNF of the ideal, and thus does not require an additional 4-dimensional lattice reduction step as was done in the implementation from [7]. A precise statement and a proof of this result are given in Appendix A.3 as Lemma 8.

Using the HNF is also convenient as it turns out to have a very specific shape that can be exploited for our analysis. Indeed, building on [21] and the generalizations in [22], we obtain that the HNF of some left $\mathcal{O}_0$-ideal have a very simple structure.

Lemma 6. *Let I be a primitive $\mathcal{O}_0$-ideal smallest in its class, then there exists $x, y < 2N$ with $x = 0 \mod 2$ and $y = 1 \mod 2$ such that I is generated by*

$$I = \langle \beta_1, \beta_2, \beta_3, \beta_4 \rangle = \left\langle N, \mathbf{i}N, \frac{x + \mathbf{i}y + \mathbf{j}}{2}, \mathbf{i}\frac{(x + \mathbf{i}y + \mathbf{j})}{2} \right\rangle$$

with $x^2 + y^2 + p = 0 \mod 4N$.

Proof. First, because I is a primitive $\mathcal{O}_0$-ideal smallest in its class, it must be, following the nomenclature of [21], inert. Indeed, as all split prime ideals are endomorphism of E_0, I having a split factor would contradict its minimality in its class. Therefore, as I is inert, there are such x, y following [23, lemma 6]. □

Sampling space for α_0. For the rest of this paper we are going to write β^I for the third element $\frac{x+\mathbf{i}y+\mathbf{j}}{2}$ of the basis given by Lemma 6 and z^I for $x + \mathbf{i}y$. From Lemma 6, we get that $\sum_{\kappa=1}^{4} c_\kappa \cdot \beta_\kappa$ is equal to $(c_1 + \mathbf{i}c_2)N + (c_3 + \mathbf{i}c_4)\beta^I$. By Remark 1, the rest of the computation, and in particular the value of z, is mostly determined by the values of c_3, c_4, and almost does not depend on the specifc values of c_1, c_2. Therefore, we choose the values of c_1, c_2 that minimize the norm of the resulting α_0.

As a consequence, α_0 is uniquely determined by c_3, c_4 and this motivates the introduction of a few additional notations. For $c = (c_3, c_4) \in \mathbb{Z}^2$, we write $z_c = c_3 + \mathbf{i}c_4$, whose norm is $n_c = \sqrt{c_3^2 + c_4^2}$. Then the α_0 associated to c is equal to $\alpha_0(c) := z_c\beta_I \mod N\mathbb{Z}[\mathbf{i}]$. $\Lambda^I(c)$ denotes the lattice $\Lambda(\alpha_0(c))$, and the first minimum of this lattice is $\lambda^I(c)$. We denote by $\omega^I(c)$ an element of $\Lambda^I(c)$ realizing this minimum, and by $\ell^I(c)$ the biggest integer such that $\omega^I(c) \in \ell^I(c)\mathbb{Z}[\mathbf{i}]$. When $c = (1, 0)$ we simply omit the c and just write $\Lambda^I, \lambda^I, \omega^I, \ell^I$.

In Proposition 2, we summarize some of the properties satisfied by α_0 thanks to this method of sampling. In particular, if (c_3, c_4) are not too big, the size condition on the norm of α_0 (which depends on the B_α from Proposition 2 below) is satisfied. Furthermore, have $c_3 \neq c_4 \mod 2$ implies that $\mathrm{tr}(\alpha) \neq \mathrm{tr}(\mathbf{i}\alpha) \mod 2$ and thus that $\alpha_0 \notin \mathbb{Z}[\mathbf{i}, \mathbf{j}]$.

Proposition 2. *Let $B_\alpha(N) \in \mathbb{N}$. Then, we can sample $\alpha_0 := \sum_{\kappa=1}^{4} c_\kappa \cdot \beta_\kappa = a_\alpha + \mathbf{i}b_\alpha + \alpha_2\mathbf{j} \notin \mathbb{Z}[\mathbf{i}, \mathbf{j}]$ such that $2(a_\alpha + b_\alpha) = 1 \mod 2$ and $2a_\alpha, 2b_\alpha \leq 2N$ and $n(\alpha_0) \leq 2N^2 + B_\alpha^2 p/4$.*

Proof. Follows directly from the above explanations, details are in Appendix A.3.

4.2 Minima of the $\Lambda^I(c)$ lattices

In this section, we study the two-dimensional lattices $\Lambda^I(c)$. We give a number of results that will prove necessary to our method of finding a good $\alpha_0(c)$.

Characterizing suitable lattices. We remind the reader that our goal is to find a c such that the second minimum of $\Lambda^I(c)$ is upper-bounded by $B_2(N)$. Henceforth, a dimension 2 lattice of determinant $4N$ satisfying this condition is called *balanced*.

Through Minkowski's second theorem we have $\lambda_2 \leq B_2(N) \Rightarrow \lambda_1 \geq 8N/(\pi B_2(N))$. We use the following Lemma 7 to characterize a sufficient condition for $\Lambda^I(c)$ to be balanced that we later use to find a good α_0.

Lemma 7. *Let Λ be a lattice of determinant $4N$, and let $B_2(N) \geq \sqrt{8N/\pi}$. If Λ contains a primitive element z of norm in $[B_1(N), B_2(N)]$ where $8N/(\pi B_2(N)) \leq B_1(N) \leq \sqrt{8N/\pi}$, then $\lambda_2(\Lambda) \leq B_2(N)$.*

Proof. By definition, if z is primitive, no $z/\ell \in \Lambda$ for any $\ell > 1$. Thus, either z is the smallest vector in the lattice and since $n(z) \geq B_1(N) \geq 8N/(\pi B_2(N))$, Minkowski's theorem prove that λ_2 must be smaller than $B_2(N)$. If z is not the smallest vector, then as z is primitive it is not a scalar multiple of the smallest vector and so $\lambda_2(\Lambda) \leq n(z) \leq B_2(N)$.

An important sub-lattice of $\Lambda^I(c)$. Our main idea to find balanced lattices $\Lambda^I(c)$ is the observation that for some good choices of c, the smallest vector of $\Lambda^I(c)$ is linked to the smallest vector of Λ^I. We use the sub-lattice of $\Lambda^I(c)$ identified in Proposition 3 below.

Proposition 3. *For any $c \in \mathbb{Z}^2$, and $\omega \in \Psi(\Lambda^I)$ there is a sub-lattice of $\Lambda^I(c)$ isometric to $z_c\omega\mathbb{Z} + 4N\mathbb{Z}[\mathbf{i}]$ through the map Ψ.*

Proof. The definition of $\Lambda^I(c)$ is the set (s,t) such that $\mathrm{tr}((s - \mathbf{i}t)z_c\beta^I) = 0 \mod 4N$. It is clear that $4N\mathbb{Z}[\mathbf{i}]$ is part of the set of solutions. Then, we just have to show that $z_c\overline{\omega}$ is part of the set of the lattice too. It follows from $\mathrm{tr}(\overline{z_c\omega}z_c\beta^I) = n_c^2\mathrm{tr}(\overline{\omega}\beta^I)$ as $\omega \in \Lambda^I$. Since Ψ is an isometry, this proves the result.

Primitive elements of $\Lambda^I(c)$. Following Proposition 3, we choose elements c such that $\omega^I(c)$ is a multiple of $z_c\omega^I$ (and often exactly equal). We are now going to state and prove a series of results to help us understand what are the conditions to make this sampling work. In particular, they help us to understand when $z_c\omega^I$ is primitive. For that, we first need one additional definition. For $0 < a < b$, we define

$$S_{\mathbf{i}}(a,b) := \left\{ n \in \mathbb{N} \;\middle|\; n \in [a^2, b^2], \text{ and } \exists c,d \in \mathbb{Z} \text{ with } n = c^2 + d^2 \text{ and } \gcd(c,d) = 1 \right\}$$

Proposition 4. *There are no $\ell' | \ell^I(c)$ such that $\gcd(\ell', 4N) = 1$.*

Proof. By absurd, consider a ℓ', divisor of $\ell^I(c)$ and coprime to $4N$. Then, $\omega^I(c)$ is in $\ell'\mathbb{Z}[\mathbf{i}]$ and if we consider $\omega^I(c)/\ell'$, we obtain

$$\ell' \cdot \mathrm{tr}((\overline{\omega^I(c)}/\ell')z_c\beta^I) = \mathrm{tr}(\overline{\omega^I(c)}z_c\beta^I) = 0 \mod 4N$$

and thus, as ℓ' is invertible, that $(\omega^I(c)/\ell') \in \Psi(\Lambda^I(c))$, breaking the minimality of $\omega^I(c)$.

Proposition 5. *For any $n_c^2 \in S_{\mathbf{i}}(a,b)$ for some $0 < a < b$, there exists a choice of element $z_c \in \mathbb{Z}[\mathbf{i}]$ of norm equal to n_c such that the biggest ℓ such that $z_c\omega^I \in \ell\mathbb{Z}[\mathbf{i}]$ is ℓ^I.*

Proof. By definition of $S_{\mathbf{i}}(a,b)$ we can choose a c such that z_c is primitive and has norm n_c. We also have that ω^I/ℓ^I must be primitive by the maximality of ℓ^I. We then want to prove the existence of c' with $n_c = n_c'$ such that that $z_{c'}\omega^I/\ell^I$ is primitive. Denote ℓ the largest integer such that $z_{c'}\omega^I \in \ell\mathbb{Z}[i]$, and let q be

any prime $q|(\ell/\ell^I)$. If $q = 3 \mod 4$, then $(z_{c'}\omega^I/\ell^I) \in q\mathbb{Z}[\mathbf{i}]$ if and only if either $z_{c'}$ or ω^I/ℓ^I are in $q\mathbb{Z}[\mathbf{i}]$, which cannot be as they are both primitive. Hence, such q cannot divide (ℓ/ℓ^I). If $q = 1 \mod 4$ or $q = 2$, then, the only solution for $(z_{c'}\omega^I/\ell^I) \in q\mathbb{Z}[\mathbf{i}]$ with primitive $z_{c'}$ and ω^I/ℓ^I is if $z_{c'} \in \mathfrak{q}$ while $(\omega^I/\ell^I) \in \overline{\mathfrak{q}}$ (where $\mathfrak{q}$ is one of the two ideals of norm q in $\mathbb{Z}[\mathbf{i}]$). In this case, it suffices to take as c' the element of norm n_c such that $z_{c'} \in \overline{\mathfrak{q}}$ by multiplying (possibly multiple times) z_c by $z_{\overline{\mathfrak{q}}}^2/q$ where $z_{\overline{\mathfrak{q}}}$ is the generator of the principal ideal $\overline{\mathfrak{q}}$. This can be done repeatedly for all prime divisors of (ℓ/ℓ^I) until the desired property is obtained.

Both of the following propositions are proven by absurd and showing a contradiction to the minimality of the lattice minima. Detailed proofs are in Appendix A.3.

Proposition 6. *Let us take an odd $n_c^2 \in S_{\mathbf{i}}(a, b)$ for some $1 < a < b$, with n_c^2 dividing ℓ^I. Let $z_c \in \mathbb{Z}[\mathbf{i}]$ be an element of norm n_c and such that ℓ^I is the biggest ℓ such that $z_c\omega^I \in \ell\mathbb{Z}[\mathbf{i}]$.*[7] *Then, $\lambda_1^I(c) = \lambda_1^I/n_c$.*

Proposition 7. *Let us take an odd $n_c^2 \in S_{\mathbf{i}}(a, b)$ for some $1 < a < b$, such that n_c^2 is coprime to ℓ^I. Let $z_c \in \mathbb{Z}[\mathbf{i}]$ be an element of norm n_c and such that ℓ^I is the biggest ℓ such that $z_c\omega^I \in \ell\mathbb{Z}[\mathbf{i}]$.*[5] *Then, $z_c\omega^I$ is a primitive element of $\Psi(\Lambda^I(c))$.*

4.3 Sampling a balanced α_0

We are now ready to state Theorem 1, our main result on the existence of a suitable α_0 with balanced lattice $\Lambda(\alpha_0)$. Following Lemma 7 and the results of Section 4.2, the idea is to choose a c such that $\Lambda^I(c)$ contains a primitive elements whose norm is in the right interval. We then prove Theorem 1 under a very plausible heuristic (stated as Assumption 1) on the existence of elements coprime to N in $S_{\mathbf{i}}(a, b)$ for large enough intervals $[a, b]$.

The proof of Theorem 1 naturally gives two algorithms that we call Loop1 and DivideMinima and which can be used to find a suitable α_0 following the theorem. Those algorithms are introduced at the end of this section, and are used in our new Qlapoty algorithm introduced in Section 5.

Assumption 1. *For any $N \in \mathbb{N}$, there exists a small constant C such that for all $0 < a < b < \sqrt{N}$, if $b - a > C \log^2 N$ then there exists $n \in S_{\mathbf{i}}(a, b)$ with* $\gcd(n, 2N) = 1$.

We believe this assumption to be quite conservative. We verified it experimentally with $C = 1$ for random values of N. Hence, we will use $C = 1$ for concrete computations. For most N the assumption would potentially remain true if the interval was linear in $\log N$ but it might not be sufficient in the worst possible case (where N is the product of the smallest primes equal to 1 mod 4). The quadratic dependency in $\log N$ is sufficient for our application to cryptography.

[7] As enabled by Proposition 5

Theorem 1. *Under Assumption 1, if we take* $B_\alpha(N) \gg \log^6 N$, $B_1(N) \leq \sqrt{8N/\pi}$ *and* $\sqrt{5}B_1(N) < B_2(N) < DB_1(N)$ *for some constant* D, *then there exists* $c = (c_3, c_4) \in [-B_\alpha(N)/2, B_\alpha(N)/2]^2$ *such that* $\lambda^I(c) \geq B_1(N)$ *if*

$$\lambda^I \geq \frac{B_1(N)}{B_\alpha(N)/\sqrt{2} - C\log^2 N} \approx \frac{\sqrt{2}B_1(N)}{B_\alpha(N)}.$$

Proof. In the case where $\lambda^I \geq B_1(N)$, the proposition is directly true, so for the rest of the proof we assume that $\lambda^I < B_1(N)$ which implies $B_1(N)/\lambda^I > 1$.

Since $B_2(N) > \sqrt{5}B_1(N)$, there exists at least one element n_c in the set $S_{\mathbf{i}}(B_1(N)/\lambda^I, B_2(N)/\lambda^I)$ as if n^2 is the biggest sum of two squares smaller than $B_1^2(N)/(\lambda^I)^2$ (it always exists as $B_1(N)/\lambda^I > 1$), then $5n^2$ will be inside the right interval and satisfy the necessary constraints. Assume that we have taken such an element n_c and let $c \in \mathbb{Z}^2$ be some element of norm n_c.

Following Proposition 3, we have that $z_c\omega^I \in \Psi(\Lambda^I(c))$. Moreover, its norm is $n_c\lambda^I$. Thus, $z_c\omega^I$ is an element of $\Psi(\Lambda^I(c))$, whose norm is in $[B_1(N), B_2(N)]$. If $z_c\omega^I$ is primitive in $\Psi(\Lambda)$, the result is obtained by Lemma 7.

We will now show that if n_c^2 is coprime to $4N$, then c can be chosen to ensure that $z_c\omega^I$ is primitive. By absurd, assume there exists $\ell > 1$ maximal such that $z_c\omega^I \in \ell\Lambda^I(c)$. This implies that:

1. $z_c\omega^I \in \ell\mathbb{Z}[\mathbf{i}]$
2. $\mathrm{tr}(\overline{z_c\omega^I}z_c\beta^I) = n_c^2\mathrm{tr}(\overline{\omega^I}\beta^I) = 0 \mod 4N\ell$.

Since n_c^2 is odd, we have, following Proposition 5, that there exists z_c' such that $n_c = n_{c'}$ and the biggest ℓ' such that $z_{c'}\omega^I \in \ell'\mathbb{Z}[\mathbf{i}]$ is ℓ^I, so we can replace c by c'. Then ℓ must divide ℓ^I by the maximality of ℓ^I. As a consequence, a prime q dividing ℓ is such that $q|\ell^I$ and so $q|4N$ by Proposition 4. If we can show that q must also divide n_c^2, we have a contradiction by the coprimality of $4N$ and n_c^2. This is what we do now. Assume that q is coprime to n_c^2. Then, as n_c^2 is invertible $\mod 4Nq$ we get:

$$n_c^2 \cdot q \cdot \mathrm{tr}((\overline{\omega^I}/q)\cdot\beta^I) = 0 \mod 4Nq \Longrightarrow \mathrm{tr}((\overline{\omega^I}/q)\cdot\beta^I) = 0 \mod 4N.$$

As q divides ℓ^I we also have that $\omega^I/q \in \mathbb{Z}[\mathbf{i}]$, which implies that $\omega^I/q \in \Lambda^I$ hence breaking the minimality of ω^I. Thus, we must have $q|n_c^2$ which contradicts our assumption that $\ell > 1$, proving the primitivity of $z_c\omega^I$.

Then, as $z_c\omega^I$ is a primitive element of $\Lambda^I(c)$ with norm contained in $[B_1(N), B_2(N)]$, the result will be proven following Lemma 7, provided we can sample z_c whose square norm n_c^2 is an element coprime to $2N$ in $S_{\mathbf{i}}\left(B_1(N)/\lambda^I, \min\left(B_2(N)/\lambda^I, B_\alpha(N)/\sqrt{2}\right)\right)$. We distinguish the 2 possible cases.

– *If* $\min\left(B_2(N)/\lambda^I, B_\alpha(N)/\sqrt{2}\right) = B_\alpha(N)/\sqrt{2}$. The condition

$$\lambda^I \geq \frac{B_1(N)}{B_\alpha(N)/\sqrt{2} - C\log^2 N}$$

then ensures that the conditions of Assumption 1 are satisfied and therefore the existence of a suitable n_c and of c satisfying all the necessary conditions.

– *If* $\min\left(B_2(N)/\lambda^I, B_\alpha(N)/\sqrt{2}\right) = B_2(N)/\lambda^I$, then it is possible that Assumption 1 cannot be used directly. In such a case, we have that

$$\frac{B_2(N) - B_1(N)}{\lambda^I} \leq C \log^2 N.$$

Following our bounds on $B_1(N)$ and $B_2(N)$, and Assumption 1, we have that:

$$(\sqrt{5}-1)\frac{B_1(N)}{\lambda^I} \leq \frac{B_2(N) - B_1(N)}{\lambda^I} \leq C \log^2 N$$

Let g be the smallest exponent such that $\sqrt{5^g} > C\log(N)^2$, and let $5^h = \gcd(5^g, \ell^I)$. Then, following Proposition 6, we can find $c_0 \in \mathbb{Z}^2$ of norm $\sqrt{5}^h$ corresponding to some $z_{c_0} \in \mathbb{Z}[\mathbf{i}]$ such that $\lambda^I(c_0) = \lambda^I/\sqrt{5^h}$.
If

$$\frac{B_2(N) - B_1(N)}{\lambda^I(c_0)} = \frac{(B_2(N) - B_1(N))\sqrt{5^h}}{\lambda^I} \geq C \log^2 N,$$

then we can reapply the same reasoning to $\Lambda_I(c_0)$ instead of Λ^I to find c_1 and z_{c_1} of square norm coprime to $2N$, and get a c corresponding to $z_c = z_{c_0} z_{c_1}$ that satisfies all the desired properties (we will verify at the end of the proof that the bound on the norm of c is still satisfied).
If we are unfortunate, and still have

$$\frac{B_2(N) - B_1(N)}{\lambda^I(c_0)} < C \log^2 N,$$

then $h < g$ by definition of the exponent g. This means in particular that $\ell^I(c_0) = \ell^I/5^h$ will be coprime to 5.
Then, because $B_2(N) > \sqrt{5}B_1(N)$, there is always an element of the form $\sqrt{5}^v$ in the set $S_{\mathbf{i}}(\sqrt{5^h}B_1(N)/\lambda_1^I, \sqrt{5^h}B_2(N)/\lambda_1^I)$. In that case, by applying Proposition 7 to the lattice $\Lambda^I(c_0)$ and $n_{c_1}^2 = 5^v$ we get an element z_{c_1}, from which we can derive a primitive element $z_c\omega^I/5^h$ of $\Psi(\Lambda^I(c))$ of norm $\lambda^I 5^{v-h} \in [B_1(N), B_2(N)]$ where $z_c = z_{c_0} z_{c_1}$.
In order to conclude that the solutions we found from $\Lambda^I(c_0)$ are valid, it suffices to prove that $5^g B_2(N)/\lambda^I < B_\alpha(N)/\sqrt{2}$. This is ensured by

$$5^g \frac{B_2(N)}{\lambda^I} \leq D 5^g \frac{B_1(N)}{\lambda^I} \leq \frac{DC\log^2 N}{\sqrt{5}-1} 5^g \leq \frac{DC^3 \log^6 N}{\sqrt{5}-1}$$

which is negligible before $B_\alpha(N)/\sqrt{2}$ by our assumption. This finishes the proof.

New algorithm following Theorem 1. We now introduce two algorithms to find α_0 following the ideas behind the proof of Theorem 1. The main algorithm is Loop1 that finds $n_c^2 \in S_i(a,b)$ and an element z_c of norm n_c with a, b ensuring $\lambda_1(c) \in [B_1(N), B_2(N)]$. To deal with the bad case where the interval is too small, we introduce another algorithm called DivideMinima whose goal is to apply

the strategy described at the end of the proof to reduce to a situation where the interval is big enough. Both these algorithms will be used in our new Qlapoty algorithm introduced in Section 5.

The coprimality condition $\gcd(\mathrm{tr}(\alpha_0), \mathrm{tr}(\mathbf{i}\alpha_0), 4N) = 1$ is guaranteed upon termination of Loop1 by selecting c_3, c_4 with different parities such that $\gcd(c_3, c_4) = 1$. Indeed, writing $\alpha_0 = (c_3 + \mathbf{i}c_4)\beta^I \mod N\mathbb{Z}[\mathbf{i}]$, and expanding the 1 and $\mathbf{i}$ components yields $(c_3 + \mathbf{i}c_4)(x + \mathbf{i}y)/2 = a_\alpha + \mathbf{i}b_\alpha$. We want to prove that $\gcd(2a_\alpha, 2b_\alpha, 4N) = 1$. Setting $g = \gcd(2a_\alpha, 2b_\alpha)$, we show that $(c_3 + \mathbf{i}c_4)(x + \mathbf{i}y) = 0 \mod g\mathbb{Z}[\mathbf{i}]$, and g^2 divides $4(a_\alpha^2 + b_\alpha^2) = (x^2 + y^2)(c_3^2 + c_4^2)$. Let q be some prime dividing both g and $4N$. First, we get that q cannot divide $x^2 + y^2$. Indeed $x^2 + y^2 = -p \mod 4N$, and since p is prime, it is invertible mod$4N$ and so we can't have any q dividing both $x^2 + y^2$ and $4N$. Thus, $(c_3 + \mathbf{i}c_4)(x + \mathbf{i}y)(x - \mathbf{i}y) = (c_3 + \mathbf{i}c_4)(x^2 + y^2) = 0 \mod q\mathbb{Z}[\mathbf{i}]$ implies that $c_3 + \mathbf{i}c_4 = 0 \mod q\mathbb{Z}[\mathbf{i}]$ by invertibility of $x^2 + y^2$ modulo q. This contradicts the fact that $\gcd(c_3, c_4) = 1$ thus proving that $\gcd(\mathrm{tr}(\alpha_0), \mathrm{tr}(\mathbf{i}\alpha_0), 4N) = 1$.

Algorithm 2 Loop1 (z^I, ω, a, b)

Require: β^I an element of I, bounds $a < b$, ω the minima of $\Lambda(\beta^I)$.

Ensure: α_0 a generator of I with its lattice $\Lambda(\alpha_0)$ such that $\lambda_1(\Lambda(\alpha_0)) \geq \sqrt{N/2}$. Also return ρ_1 and ρ_2, solution of $\mathrm{tr}(\overline{\rho}\alpha_0) = 1 \mod N$

$\ell \leftarrow \gcd(s_0, t_0)$ ▷ $s_0 + \mathbf{i}t_0 \leftarrow \omega$
while True **do** ▷ Loop 1
  if b < 5 **then** ▷ ensure correct sampling when $\sqrt{b} - \sqrt{a} < 1$.
    $c_3 \leftarrow 1$, $c_4 \leftarrow 0$
  else
    Select a random $c_3 \in \mathbb{Z} \cap [\sqrt{a/2}, \sqrt{b}]$
    Select a random $c_4 \in \mathbb{Z} \cap [\sqrt{\max(0, a - c_4^2)}, \sqrt{b - c_4^2}]$
  end if
  if c_3 and c_4 have same parity: **continue**
  if $\gcd(c_3, c_4) \neq 1$: **continue**
  if $\sqrt{b} - \sqrt{a} \geq \log(N)^2$ **then**
    if $\gcd(c_3^2 + c_4^2, N) \neq 1$: **continue**
  else
    if $\gcd(c_3^2 + c_4^2, \ell) \neq 1$: **continue**
  end if
  $z_c \leftarrow c_3 + \mathbf{i}c_4$
  while $\gcd_{\mathbb{Z}[\mathbf{i}]}(\overline{z_c}, \omega) \notin \{\pm 1, \pm\mathbf{i}\}$ **do**
    $\gamma \leftarrow \gcd_{\mathbb{Z}[\mathbf{i}]}(\overline{z_c}, \omega)$
    $z_c \leftarrow z_c \cdot \gamma^2 / n(\gamma)$ ▷ $n(\gamma) = \gamma\overline{\gamma}$
  end while
  $\alpha_0 \leftarrow z_c \cdot \beta^I \mod N\mathbb{Z}[\mathbf{i}]$
  $\Lambda(\alpha_0), (\rho_1, \rho_2) \leftarrow$ ComputeLattice(α_0) ▷ $\Lambda(\alpha_0)$ unreduced
  $\omega \leftarrow z_c \cdot \omega$
  $\Lambda(\alpha_0) \leftarrow$ ReduceGivenMinima$(\Lambda(\alpha_0), \omega)$ ▷ $\Lambda(\alpha_0)$ reduced
  return $\alpha_0, \Lambda(\alpha_0), \rho_1, \rho_2$.
end while

Algorithm 3 DivideMinima (z^I, ω, a, b)

Require: $\beta^I \in \mathcal{O}_0$ the "canonical" second generator of I given by the 3rd row of $\mathsf{HNF}(I)$, ω the first minimum of $\Lambda(\beta^I)$, bounds $a < b$

Ensure: Updated β^I, ω and a, b such that $\omega \notin 5\mathbb{Z}[\mathbf{i}]$ and still realise the minima of $\Lambda(\beta^I)$.

$s_0 + \mathbf{i}t_0 \leftarrow \omega$
$\ell^I \leftarrow \gcd(s_0, t_0)$
Let h be the smallest integer such that $5^h > \log^4 N$.
$5^h \leftarrow \gcd(5^h, \ell^I)$
$s_0 + \mathbf{i}t_0 \leftarrow (s_0 + \mathbf{i}t_0)/5^h$
if $2s_0 - t_0 \neq 0 \mod 5$ **then** ▷ $2 - \mathbf{i}$ does not divide $s_0 + \mathbf{i}t_0$
    $\omega \leftarrow (2 + \mathbf{i})^h \cdot (s_0 + \mathbf{i}t_0)$, $\beta^I \leftarrow (2 + \mathbf{i})^h \cdot \beta^I$
else
    $\omega \leftarrow (2 - \mathbf{i})^h \cdot (s_0 + \mathbf{i}t_0)$, $\beta^I \leftarrow (2 - \mathbf{i})^h \cdot \beta^I$
end if
$a \leftarrow 5^h \cdot a, b \leftarrow 5^h \cdot b$ ▷ $\lambda_1\Big(\Lambda\big((2 \pm \mathbf{i})^h \cdot \beta^I\big)\Big) = \lambda_1\big(\Lambda(\beta^I)\big)/5^h$
return β^I, ω, a, b.

4.4 Analyzing the failure probability of the new algorithm to sample α_0.

We are now going to identify the failure cases for our Loop1 algorithm. Write $B(p, N)$ for the bound on λ^I given by Theorem 1:

$$B(p, N) := \frac{B_1(N)}{B_\alpha(N)/\sqrt{2} - C\log^2 N}.$$

We say that this bound depends on p as we set $B_\alpha(N) = \sqrt{\frac{2^{e+3}N - 16N^2}{p}}$ in our Qlapoty algorithm.

By Theorem 1 the existence of a suitable output for the first loop is not guaranteed if the smallest ideal I (of norm N) in the class of the input ideal J is such that $\lambda^I \leq B(p, N)$. The additional condition $B_\alpha(N) \gg \log^6 N$ from Theorem 1 requires to have N not too small (as we have $B_\alpha(N) \sim \sqrt{N}$). To ensure that we are in a regime where that condition can be safely assumed, we need $N > p^{1/4}$. This bound is chosen so that the proportion of ideals not meeting this requirement is smaller than the proportion of ideals not meeting the requirement on the first minimum (see the heuristic estimate at the end of this section).

When I satisfies any of the two conditions above, we say that I is a **failing ideal**, or that J is in a **failing ideal class** if the smallest I equivalent to J is a failing ideal.

When J belongs to a uniformly random ideal class, we write $\mathrm{P}_{\mathsf{fail\text{-}ideal}}$ for the probability that this is a failing ideal class. The goal of the rest of this section is to provide a bound on $\mathrm{P}_{\mathsf{fail\text{-}ideal}}$.

Remark 5. As we aim at providing a strict and credible upper-bound on the failure probability, our definition of failing ideals is deliberately too strict. It is very likely that our algorithm would succeed for some ideals that meet this definition (in particular ideals with a small norm N).

An exact bound on $\mathbf{P}_{\mathsf{fail\text{-}ideal}}$. Our final result on $\mathrm{P}_{\mathsf{fail\text{-}ideal}}$ is given in Proposition 9. But first, we need some preliminary results. With Proposition 8 below, we give a set of necessary conditions for an ideal I to be failing.

Proposition 8. *If $I = \mathcal{O}_0\langle N, \frac{x+\mathbf{i}y+\mathbf{j}}{2}\rangle$ of norm N is a failing ideal because λ^I is too small, then there exists $g, s_0, t_0, \lambda \in \mathbb{Z}$ such that $g \leq B(p,N)$ and $4N = 0 \mod g$ and $s_0^2 + t_0^2 \leq B(p,N)^2$ as well as:*

- *Either $s_0 \neq 0$ and then:* $\gcd(\lambda, 4N/g) = 1$ *and* $\lambda(y,x) = (s_0, t_0) \mod (4N/g)$ *with* $\lambda^2 = -(s_0^2 + t_0^2)/p \mod (4N/g)$.
- *or $s_0 = 0$ and $t_0 = g$ and $y = 0 \mod 4N/g$ and $N = 0 \mod 4N/g$.*

Proof. Since I is the smallest ideal in its class, following Lemma 6, there always exists $0 \leq x, y < 2N$ such that $I = \mathcal{O}_0\langle N, \frac{x+\mathbf{i}y+\mathbf{j}}{2}\rangle$, and we have $x^2 + y^2 = -p \mod 4N$.

The lattice Λ^I corresponds to the set of elements (s,t) such that $xs + ty = 0 \mod 4N$. If $y = 0$, then it is clear that the vector $[0,1]$ is in Λ^I and so we can take $(s_0, t_0) = (0,1)$ and $g = 1$. If $y \neq 0$, then let us write $g_y = \gcd(4N, y)$, $y' = y/g_y$. Let (s_0, t_0) be an element realizing the first minimum of Λ^I.

If I is a failing ideal because λ^I is too small, then $s_0^2 + t_0^2$ is smaller than $B(p,N)^2$ which implies that $s_0 \leq B(p,N)$.

Let us assume that $s_0 \neq 0$ and take $g_0 = \gcd(s_0, 4N)$. By the basis $\langle (g_y, c_y), (0, 4N/g_y)\rangle$ of Λ^I given in Lemma 5 with $c_y = x/y' \mod 4N/g_y$, we get:

1. g_y must divide s_0 which implies that g_y divides g_0, and that $4N/g_0$ divides $4N/g_y$. Moreover, y' is also invertible $\mod 4N/g_0$ and the equality $c_y = x/y' \mod 4N/g_0$ is true.
2. there exists a coefficient $s_1 \neq 0$ such that $s_0 = s_1 g_0 = s_1(g_0/g_y)g_y$ and $t_0 = s_1(g_0/g_y)c_y \mod (4N/g_y)$.

This implies that there exists some $s_0', t_0' \in \mathbb{Z}$ and some $s_2 \in \mathbb{Z}$ such that

$$(s_0, t_0) = \frac{g_0}{g_y}(s_1 g_y, s_1 c_y + s_2 \frac{4N}{g_0}) = \frac{g_0}{g_y}(s_0', t_0')$$

By definition, s_1 is also invertible modulo $4N/g_0$. Thus, by setting $\lambda = s_1/y' \mod (4N/g_0)$ we get

$$(s_0', t_0') = \lambda(y,x) \mod (4N/g_0)$$

By taking $g = g_0$ and replacing (s_0, t_0) by (s_0', t_0'), all the conditions of the proposition are satisfied.

If $s_0 = 0$, then $yt_0 = 0 \mod 4N$ so $t_0 = 4N/g_y$. In that case, we can take $g = 4N/g_y$. Then $g = t_0 \leq B(p,N)$ as $t_0^2 + s_0^2 = t_0^2 \leq B(p,N)^2$. Clearly, $4N = 0 \mod g$ and then $y = 0 \mod 4N/g$ as $4N/g = g_y$. Finally, since y is odd by Lemma 6 we have that $\gcd(y, 4N) = \gcd(y, N)$, and so we get $N = 0 \mod 4N/g$ too.

Using the conditions listed in Proposition 8, we can give an upper-bound on the probability that a random ideal class is a failing class.

Proposition 9. *The probability $P_{\text{fail-ideal}}$ that a uniformly random ideal class is a failing class satisfies (for $h(-p)$ denoting the class number of $\mathbb{Z}[\frac{1+\sqrt{-p}}{2}]$):*

$$\frac{p}{12} P_{\text{fail-ideal}} \leq \frac{1}{4}\sqrt{p}(\log p + 8) + \frac{1}{8}h(-p) \max_{p^{1/4} \leq N \leq \sqrt{p}/2} B(p,N)^2 \log(B(p,N)) + \sum_{p^{1/4} \leq N \leq \frac{\sqrt{p}}{2}} \sum_{\substack{g \mid 4N \\ g \leq B(p,N)}} \frac{g^2}{2} \sum_{s_0^2 + t_0^2 \leq B(p,N)^2} \#\{\lambda \in \mathbb{Z}/\frac{4N}{g}\mathbb{Z}^\times \mid \lambda^2 = -\frac{s_0^2 + t_0^2}{p}\}$$

Proof. As there are $\approx p/12$ different ideal classes, we can upper-bound $\frac{p}{12}\mathrm{P}_{\text{fail-ideal}}$ by the number of failing ideal classes.

Let us take a random ideal class, and let I be the smallest ideal in this class. Then, we know that $N \leq \frac{\sqrt{p}}{2}$. We are going to upper bound the number of failing ideal classes by the number of failing ideals[8] whose norm is smaller than $\frac{\sqrt{p}}{2}$. First, we need to take into account all the ideals whose norm N satisfies $N < p^{1/4}$. By [2, Lemma 5.7] there are at most $\frac{1}{4}\sqrt{p}(\log p + 8)$ such ideal classes.

We write $I = \mathcal{O}_0\langle N, \frac{x+\mathbf{i}y+\mathbf{j}}{2}\rangle$ and assume that $p^{1/4} \leq N \leq \frac{\sqrt{p}}{2}$ from now on. Such an ideal I can only be a failing ideal if the condition on the first minimum λ^I is not met. Lets take s_0, t_0, g, λ as provided by Proposition 8 in this case.

When $s_0 \neq 0$, the element λ is invertible $\mod 4N/g$, and so these conditions imply that $(y, x) = \lambda^{-1}(s_0, t_0) \mod 4N/g$. Since $0 \leq x, y < 2N$, there are at most $g^2/4$ different possible cpairs x, y satisyfing this equation mod $4N/g$ for a given tuple g, s_0, t_0, λ.

Thus, we get the following upper-bound on the number of failing ideal for which $s_0 \neq 0$:

$$\sum_{p^{1/4} \leq N \leq \frac{\sqrt{p}}{2}} \sum_{\substack{g \mid 4N \\ g \leq B(p,N)}} \frac{g^2}{2} \sum_{s_0^2 + t_0^2 \leq B(p,N)^2} \#\{\lambda \in \mathbb{Z}/\frac{4N}{g}\mathbb{Z}^\times \mid \lambda^2 = -\frac{s_0^2 + t_0^2}{p}\}$$

If $s_0 = 0$, then $y = 0 \mod 4N/g$ which implies that $\frac{x+\mathbf{i}y+\mathbf{j}}{2} = \frac{x+z'\mathbf{j}}{2} + y'(4N/g)\frac{\mathbf{i}+\mathbf{j}}{2} \in I$ with $z' = 1 - y'(4N/g)$. This means that:

$$I + \mathcal{O}_0\frac{4N}{g} = \mathcal{O}_0\frac{x + z'\mathbf{j}}{2} + \mathcal{O}_0\frac{4N}{g}$$

[8] left $\mathcal{O}_0$ primitive ideals

Ideals of the form $\mathcal{O}_0\frac{a+bj}{2} + \mathcal{O}_0 M$ can be rewritten as $\mathcal{O}_0\mathfrak{a}$ where $\mathfrak{a}$ is an image of an ideal of the imaginary quadratic order $\mathbb{Z}[\frac{1+\sqrt{-p}}{2}]$ through the natural embedding of this order inside $\mathcal{B}_{p,\infty}$. There can be only be at most $2h(-p)$ distinct classes of such ideals by [13]. Since $4N/g$ divide N, this means that $I \subset I + \mathcal{O}_0(4N/g)$, and so I can be factored as $(I + \mathcal{O}_0(4N/g)) \cdot J$ where J is a cyclic ideal of norm $g/4$. (Here J is an ideal in a maximal order of $\mathcal{B}_{p,\infty}$ not necessarily $\mathcal{O}_0$.) Using again [2, Lemma 5.7] to count how many ideals of norm smaller than $B(p,N)/4$ there are, we get an upper bound of $\frac{1}{8}h(-p)\max_{p^{1/4}\leq N\leq \frac{\sqrt{p}}{2}} B(p,N)^2 \log B(p,N)$ distinct failing ideals for which $s_0 = 0$ and $p^{1/4} \leq N \leq \frac{1}{2}\sqrt{p}$.

Comparing Proposition 9 with the state of the art analysis of Qlapoti. Our analysis refutes the analysis of the failure of Qlapoti provided in [7, Section 4]. The error made there is to heuristically assume that the lattices $\Lambda(\alpha_0)$ behave like random lattices. We have exhibited that there is a strong link between $\Lambda(\beta^I)$ and $\Lambda(z_c\beta^I)$ when z_c is not too big, which is the case in Qlapoti. As such, when $\Lambda(\beta^I)$ contains a very small element, the Qlapoti algorithm cannot succeed. This explains for instance the failure observed when I corresponds to an isogeny defined over $\mathbb{F}_p$ as $\Lambda(\beta^I)$ contains an element of norm 1 in that case (and why the authors of qt-pegasis [9] modified the algorithm to handle that special case).

Estimating heuristically the bound from Proposition 9. In order to simplify the Proposition 9, and ease its computation, we make a few heuristics. Mainly, for any integer M, the expected number of square roots for a random $n_0 \in \mathbb{Z}/M\mathbb{Z}$ is 1. Thus, the expected value of $\sum_{n_0\in S} \#\{\lambda \in \mathbb{Z}/M\mathbb{Z} \mid \lambda^2 = n_0\}$ when S is a uniformly random subset of $\mathbb{Z}/M\mathbb{Z}$ is $\#S$. And so assuming that the set of n_0 appearing in the sum from Proposition 9 behave like a random set, we reexpress $\mathrm{P}_{\mathsf{fail\text{-}ideal}}$ as

$$\mathrm{P}_{\mathsf{fail\text{-}ideal}} \leq \frac{1}{\sqrt{p}}\left(3(\log p + 8) + 12f(2f + \frac{1}{2}) + \frac{6}{\pi}f\log(4\sqrt{f})\log p\right). \quad (8)$$

For the full details on this simplification we refer to Appendix A.4.

5 The Qlapoty algorithm

We are finally ready to detail our new Qlapoty algorithm. It crucially relies on the building blocks introduced in Section 3 and Section 4. In Section 5.1 we detail the algorithm and a few additional tweaks, and in Section 5.2 we analyze its failure probability.

5.1 A description of the algorithm.

Our Qlapoty algorithm follows the strategy underlying the proof of Theorem 1 to sample a correct α_0 using the Loop1 and DivideMinima algorithms. We thus multiply $\beta^I = \frac{x+\mathbf{i}y+\mathbf{j}}{2}$ by z_c of norm in $[B_1(N)/\lambda^I, B_2(N)/\lambda^I]$ which suffices according to Lemma 7. In practice, we propose to use a slightly smaller interval $[B_1(N)/\lambda^I, B_M(N)/\lambda^I]$ where $B_M(N)$ is set to $\sqrt{5N/2}$. As $5/2 < 8/\pi$, Minkowski theorem implies that $\omega = z_c\omega^I$ is the smallest element of $\Lambda^I(c)$ (since we use $B_1(N) = \sqrt{N/2}$ in Qlapoty Theorem 1 still applies). This modification guarantees that a Minkowski-reduced basis of $\Lambda^I(c)$ can be computed more efficiently using the ReduceGivenMinima algorithm.

Algorithm 4 Qlapoty

Require: A left $\mathcal{O}_0$ ideal J.
Ensure: A solution β_1, β_2, θ of the norm equation satisfying the geometric conditions Section 3.1.

$I \leftarrow \mathsf{SmallestEquivIdeal}(J)$, $N \leftarrow n(I)$
$x, y \leftarrow \mathsf{HNF}(I)$ ▷ Hermite normal form, see Lemma 6
$\beta^I \leftarrow (x + \mathbf{i}y + \mathbf{j})/2$ ▷ $I = \mathcal{O}_0\langle N, \frac{x+\mathbf{i}y+\mathbf{j}}{2}\rangle$
$\Lambda(\beta^I), - \leftarrow \mathsf{ComputeLattice}(\beta^I)$
Reduce the lattice $\Lambda(\beta^I)$ ▷ Using for example Lagrange's algorithm
Let $(s_0, t_0) \in \Lambda(\beta^I)$ realize the first minimum.
$\omega \leftarrow (s_0 + \mathbf{i}t_0)$, $\lambda^I \leftarrow (s_0)^2 + (t_0)^2$. ▷ λ^I is the square of the minimum
$B_M^2(N) \leftarrow 5N/2$
$B_1^2(N) \leftarrow N/2$ ▷ Need $\sqrt{5}B_1 \leq B_M \leq DB_1$
$B_\alpha^2(N) \leftarrow N \cdot (2^{e+3} - 16N)/p$
if $\lambda^I \leq 2B_1^2(N)/B_\alpha^2(N)$: **return** $\perp$ ▷ Follows from Theorem 1
$a \leftarrow B_1^2(N)/\lambda^I$
$b \leftarrow \min\left(B_\alpha^2(N)/2, B_M^2(N)/\lambda^I\right)$
if $\sqrt{b} - \sqrt{a} \leq \log^2(N)$ **then**
    $\beta^I, \omega, a, b \leftarrow \mathsf{DivideMinima}(\beta^I, \omega, N, a, b)$
end if
while true do
    $\alpha_0, \Lambda(\alpha_0), \rho_1, \rho_2 \leftarrow \mathsf{Loop1}(\beta^I, \omega, N, a, b)$
    $\mathsf{sol} \leftarrow \mathsf{Loop2}(\alpha_0, \Lambda(\alpha_0), \rho_1, \rho_2, N)$
    if $\mathsf{sol} \neq \perp$: **return** sol.
end while

5.2 Failure Probability of Qlapoty

We now compute the failure probability of Qlapoty. We rely on the conclusion of Section 3.3 on the failure probability of the Loop2 algorithm that we recall here.

For a given α_0 the number of possible odd candidates λ is $B_\lambda/2 = \sqrt{2^{e-4}/r_0}$ where $r_0 = n(\alpha_0)/N$. Following Eq. (7), for a given λ, ρ_λ the success probability

of Loop2 is lower-bounded by

$$\rho_\lambda \geq \frac{1}{4\pi\delta^2 f \log(\sqrt{p}/(2f\delta))}.$$

Theorem 2. *For J an ideal from an uniformly random ideal class, the probability that* Qlapoty *fails on input J is smaller than $P_{\text{fail-ideal}} + p^{-1}$.*

Proof. Let J an ideal in a uniformly distributed class, and assume it is not in a failing ideal class. Let $G_{\alpha_0}(J)$ denote the set of good outputs α_0 from Loop1, with $|G_{\alpha_0}(J)| \geq 1$ by Theorem 1. Since we test $B_\lambda(\alpha_0)/2 \geq 1$ values of λ per α_0, we need

$$\sum_{\alpha_0 \in G_{\alpha_0}(J)} B_\lambda(\alpha_0)/2 \geq \rho_\lambda^{-1} \log(p) \tag{9}$$

in order to get $\Pr[\text{Qlapoty}(J) \to \perp] \leq (1-\rho_\lambda)^{\rho_\lambda^{-1}\log(p)} \leq p^{-1}$. So we now try to see if we can meet that condition, or how close we can get to it. As J is not a failing ideal class, we have that $p^{1/4} \leq N$. By Assumption 1 (and assuming $C = 1$), the size of G_{α_0} is lower-bounded by

$$|G_{\alpha_0}(J)| \geq \frac{\min\left\{B_\alpha/\sqrt{2}, B_M(N)/\lambda^I\right\} - B_1(N)/\lambda^I}{\log(N)^2}.$$

We distinguish two subcases:

1. If $B_M(N)/\lambda^I \leq B_\alpha(N)/\sqrt{2}$, then $|G_{\alpha_0}| \geq \frac{\sqrt{N}}{2\lambda^I \log(N)^2}$, since $B_M(N) = \sqrt{5N/2}$ and $B_1(N) = \sqrt{N/2}$. By Proposition 2 used with $B_M(N)/\lambda^I$:

   $$B_\lambda(\alpha_0)^2 = 2^{e-2}/r_0 \geq \frac{p}{16f\left(2N + \frac{B_M^2(N)p}{4N(\lambda^I)^2}\right)} \geq \frac{p(\lambda^I)^2}{16f(128N^2+5p)} \geq \frac{(\lambda^I)^2}{16f(128\delta^2+5)}.$$

   Thus, with the lower-bound $\sum_{\alpha_0 \in G_{\alpha_0}(J)} B_\lambda(\alpha_0) \geq |G_{\alpha_0}(J)| \min_{\alpha_0} B_\lambda(\alpha_0)$ and an upper bound on $\rho_\lambda^{-1}\log(p)$, we obtain that Eq. (9) holds at least if

   $$\frac{\sqrt{N}}{8\log(N)^2\sqrt{f(128\delta^2+5)}} \geq 4\pi\delta^2 f \log(p)^2$$

   which then yields after replacing N by $\log p^\chi$:

   $$\log(p)^{\chi/2-2} \geq 8\pi\delta^2 \underbrace{4\sqrt{(128\delta^2+5)}}_{B_1} \log\log(p)^2 f^{1.5}\chi^2. \tag{10}$$

2. If $B_M(N)/\lambda^I > B_\alpha(N)/\sqrt{2}$, J not failing ensures that $(\lambda^I)^2 \geq f/2$ and in fact $(f+1)/2$ as f odd and $(\lambda^I)^2 \in \mathbb{N}$. Then, by Taylor's approximation,

   $$|G_{\alpha_0}| \geq \frac{\sqrt{N}}{\sqrt{2}\log(N)^2}\left(\frac{\sqrt{2^{e+3}-16N}}{\sqrt{p}} - \frac{1}{\lambda^I}\right) \geq \frac{\sqrt{N}}{2\sqrt{f}\log(N)^2}(1-\sqrt{f/(f+1)})$$

Using the lower bound $B_\lambda/2 = 1$, Eq. (9) is satisfied if

$$\frac{\sqrt{N}}{\sqrt{f}\log(N)^2}(1-\sqrt{f/(f+1)}) \geq 8\pi\delta^2 f \log(p)^2,$$

which, by again replacing N by $\log p^\chi$, simplifies to:

$$\log(p)^{\chi/2-2} \geq 8\pi\delta^2 \underbrace{(1-\sqrt{f/(f+1)})^{-1}}_{B_2} \log\log(p)^2 f^{1.5}\chi^2. \tag{11}$$

Setting $\delta = \log(p)^\chi/\sqrt{p}$, it suffices to check that χ is a solution of Eq. (12) to ensure both Eq. (10) and Eq. (11) and hence Eq. (9):

$$p\log(p)^{-2-1.5\chi} \geq 8\pi \max\{B_1, B_2\} \log\log(p)^2 f^{1.5}\chi^2. \tag{12}$$

All χ such that $\log(p)^\chi \in [p^{1/4}, p^{1/2}]$ satisfy Eq. (12) for p larger than 200 bits. □

Following Theorem 2 and relying on Eq. (8) we upperbounds in Table 1 the failure probability of Qlapoty across SQIsign NIST 2 parameters.

Table 1. Failure probability of Qlapoty across SQIsign NIST 2 parameters.

NIST Level	$\log(p)$	f	$P_{\text{fail-ideal}}$
I	250	5	2^{-112}
III	382	65	2^{-172}
V	505	27	2^{-233}

6 Implementation and timings

We implemented Qlapoty in both SageMath and C, and the code is available at:

https://github.com/qlapoty/Qlapoty

The second implementation is based on the C implementation of Qlapoti [7] which itself is based on the round 2 NIST submission of SQIsign [1], and directly integrates Qlapoty in an implementation of SQIsign. As a consequence, it relies on GMP for integer arithmetic. Comparing our C implementation to the one of Qlapoti, we obtain Table 2 for the speed of the norm equation alone.

We observe that Qlapoty provides a significant speedup over Qlapoti, in addition to offering desirable geometric properties and an improved failure analysis.

While integrating Qlapoty in SQIsign's NIST 2 implementation using the code provided by Qlapoti, we noticed that the implementation of the ideal-to-isogeny translation was suboptimal, since it used Weil pairings to recover the

Table 2. Average C runtime of Qlapoti and Qlapoty norm equation solving across SQIsign parameter sets (10k iterations), on an intel Ultra 7 265 CPU. Timings in Mcycles.

NIST Level	e	Qlapoti	Qlapoty	factor
I	246	20.0	3.46	5.78×
III	374	148	16.1	9.2×
V	498	118	13.4	8.82×

correct isogeny, which can be fixed by the $(2, 2)$-isogeny formulae. We simplified this procedure. As visible in Table 3 and Table 4, these two changes result in significant speedups, in particular at higher security levels where the norm equation solving contributes more to the overall runtime due to the larger cofactor f. For comparison, we also provide timings of SQIsign [1] in Table 5.

The main difference between the description of Qlapoty in the above sections and the implemented version is that our implementation uses trial division to solve sum-of-two squares instances for some non-primes. This can only increase the success probability of the Cornacchia step in Algorithm 1, and therefore does not invalidate our analysis. We mostly used this in order to rely on the same subroutines as Qlapoti's C implementation.

Remark 6. We use the names "NIST lvl x" for the parameter sets of SQIsign's round 2 NIST submission. It is currently debated, following [28], whether these still meet the security requirements of the NIST levels they are named after. Nevertheless, our improvements and analysis are generic and apply to any choice and size of parameters.

Table 3. Average runtime of SQIsign signing and keygen using Qlapoti and Qlapoty norm equation solving across SQIsign parameter sets for the optimized (broadwell) implementation (1000 iterations), on an intel Ultra 7 265 CPU. Timings in Mcycles.

Parameter set	Keygen			Signing		
(NIST Level)	Qlapoti	Qlapoty	*Gain*	Qlapoti	Qlapoty	*Gain*
I	42.4	26.9	1.58×	113	82.0	1.38×
III	226	86.6	2.61×	532	245	2.17×
V	279	144	1.94×	723	446	1.62×

References

1. Aardal, M.A., Adj, G., Aranha, D.F., Basso, A., Canales Martínez, I.A., Chávez-Saab, J., Santos, M.C., Dartois, P., De Feo, L., Duparc, M., Eriksen, J.K., Fouotsa, T.B., Filho, D.L.G., Hess, B., Kohel, D., Leroux, A., Longa, P., Maino, L., Meyer,

Table 4. Average runtime of SQIsign signing and keygen using Qlapoti and Qlapoty norm equation solving across SQIsign parameter sets for the reference implementation (1000 iterations), on an intel Ultra 7 265 CPU. Timings in Mcycles.

Parameter set	Keygen			Signing		
(NIST Level)	Qlapoti	Qlapoty	*Gain*	Qlapoti	Qlapoty	*Gain*
I	68.2	46.8	1.46×	172	128	1.34×
III	279	126	2.22×	635	338	1.88×
V	396	235	1.68×	989	667	1.48×

Table 5. Average runtime of signing and keygen for unmodified the round 2 NIST submission of SQIsign [1] for the both reference and optimized implementations (1000 iterations), on an intel Ultra 7 265 CPU. Timings in Mcycles.

Parameter set	broadwell		Reference	
(NIST Level)	Keygen	Signing	Keygen	Signing
I	77.7	183	116	270
III	228	527	308	699
V	362	901	538	1274

M., Nakagawa, K., Onuki, H., Panny, L., Patranabis, S., Petit, C., Pope, G., Reijnders, K., Robert, D., Rodríguez Henríquez, F., Schaeffler, S., Wesolowski, B.: SQIsign. Tech. rep., National Institute of Standards and Technology (2024), available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures

2. Aardal, M.A., Basso, A., De Feo, L., Patranabis, S., Wesolowski, B.: A complete security proof of SQIsign. In: Kalai, Y.T., Kamara, S.F. (eds.) CRYPTO 2025, Part VI. LNCS, vol. 16005, pp. 190–222. Springer, Cham (Aug 2025). https://doi.org/10.1007/978-3-032-01887-8_7
3. Alagic, G., Bros, M., Ciadoux, P., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Liu, Y.K., Miller, C., Moody, D., Peralta, R., Perlner, R., Robinson, A., Silberg, H., Smith-Tone, D., Waller, N.: Status Report on the Second Round of the Additional Digital Signature Schemes for the NIST Post-Quantum Cryptography Standardization Process. https://csrc.nist.gov/pubs/ir/8610/final accessed on August 12, 2026 (2026)
4. Babai, L.: On Lovász' lattice reduction and the nearest lattice point problem. Combinatorica **6**(1), 1–13 (1986). https://doi.org/10.1007/BF02579403
5. Basso, A., Borin, G., Castryck, W., Corte-Real Santos, M., Invernizzi, R., Leroux, A., Maino, L., Vercauteren, F., Wesolowski, B.: PRISM: Simple and compact identification and signatures from large prime degree isogenies. In: Jager, T., Pan, J. (eds.) PKC 2025, Part III. LNCS, vol. 15676, pp. 300–332. Springer, Cham (May 2025). https://doi.org/10.1007/978-3-031-91826-1_10
6. Basso, A., Dartois, P., De Feo, L., Leroux, A., Maino, L., Pope, G., Robert, D., Wesolowski, B.: SQIsign2D-West - the fast, the small, and the safer. In: Chung, K.M., Sasaki, Y. (eds.) ASIACRYPT 2024, Part III. LNCS, vol. 15486, pp. 339–370. Springer, Singapore (Dec 2024). https://doi.org/10.1007/978-981-96-0891-1_11
7. Borin, G., Corte-Real Santos, M., Eriksen, J.K., Invernizzi, R., Mula, M., Schaeffler, S., Vercauteren, F.: Qlapoti: Simple and efficient translation of quaternion ideals to isogenies. In: Hanaoka, G., Yang, B.Y. (eds.) ASIACRYPT 2025,

Part IV. LNCS, vol. 16248, pp. 174–205. Springer, Singapore (Dec 2025). https://doi.org/10.1007/978-981-95-5113-2_6
8. Chavez-Saab, J., Santos, M.C., De Feo, L., Eriksen, J.K., Hess, B., Kohel, D., Leroux, A., Longa, P., Meyer, M., Panny, L., Patranabis, S., Petit, C., Rodríguez Henríquez, F., Schaeffler, S., Wesolowski, B.: SQIsign. Tech. rep., National Institute of Standards and Technology (2023), available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures
9. Dartois, P., Eriksen, J.K., Invernizzi, R., Vercauteren, F.: qt-Pegasis: Simpler and Faster Effective Class Group Actions. In: Daemen, J., Thomé, E. (eds.) EUROCRYPT 2026, Part IV. LNCS, vol. 16545, pp. 636–665. Springer, Cham (May 2026). https://doi.org/10.1007/978-3-032-25327-9_22
10. Dartois, P., Leroux, A., Robert, D., Wesolowski, B.: SQIsignHD: New dimensions in cryptography. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part I. LNCS, vol. 14651, pp. 3–32. Springer, Cham (May 2024). https://doi.org/10.1007/978-3-031-58716-0_1
11. 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, Part I. LNCS, vol. 12491, pp. 64–93. Springer, Cham (Dec 2020). https://doi.org/10.1007/978-3-030-64837-4_3
12. 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.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 659–690. Springer, Cham (Apr 2023). https://doi.org/10.1007/978-3-031-30589-4_23
13. Delfs, C., Galbraith, S.D.: Computing isogenies between supersingular elliptic curves over F p. Designs, Codes and Cryptography **78**(2), 425–440 (2016)
14. Deuring, M.: Die Typen der Multiplikatorenringe elliptischer Funktionenkörper. In: Abhandlungen aus dem mathematischen Seminar der Universität Hamburg. vol. 14, pp. 197–272. Springer Berlin/Heidelberg (1941)
15. Duparc, M., Fouotsa, T.B.: SQIPrime: A dimension 2 variant of SQISignHD with non-smooth challenge isogenies. In: Chung, K.M., Sasaki, Y. (eds.) ASIACRYPT 2024, Part III. LNCS, vol. 15486, pp. 396–429. Springer, Singapore (Dec 2024). https://doi.org/10.1007/978-981-96-0891-1_13
16. Hardy, G.H., Wright, E.M.: An introduction to the theory of numbers. Oxford university press (1979)
17. Johnston, D., Ramaré, O., Trudgian, T.: An explicit upper bound for L (1, χ) when χ is quadratic. Research in Number Theory **9**(4), 72 (2023). https://doi.org/10.1007/s40993-023-00476-4
18. Kani, E.: The number of curves of genus two with elliptic differentials. Journal für die reine und angewandte Mathematik **1997**(485), 93–122 (1997). https://doi.org/10.1515/1997.485.932
19. Kohel, D., Lauter, K., Petit, C., Tignol, J.P.: On the quaternion-isogeny path problem. LMS Journal of Computation and Mathematics **17**(A), 418–432 (2014). https://doi.org/10.1112/S1461157014000151
20. Lai, Y.F.: Qlapoti+ and More: Optimizing Isogeny-based Signatures. Cryptology ePrint Archive, Paper 2026/1640 (2026), https://eprint.iacr.org/2026/1640
21. Leroux, A.: Efficient quaternion algorithms for the deuring correspondence and application to the evaluation of modular polynomials. Cryptology ePrint Archive, Paper 2026/185 (2026), https://eprint.iacr.org/2026/185
22. Leroux, A., Schaeffler, S.: New algorithms for quaternion ideals in SQIsign. to appear (2026)

23. Leroux, A., Schaeffler, S.: New algorithms for quaternion ideals in SQIsign. Cryptology ePrint Archive, Paper 2026/2153 (2026), https://eprint.iacr.org/2026/2153
24. Morain, F., Nicolas, J.L.: On Cornacchia's algorithm for solving the diophantine equation $u^2 + dv^2 = m$ (1990), http://www.lix.polytechnique.fr/Labo/Francois.Morain/Articles/cornac.pdf
25. Nakagawa, K., Onuki, H., Castryck, W., Chen, M., Invernizzi, R., Lorenzon, G., Vercauteren, F.: SQIsign2D-East: A new signature scheme using 2-dimensional isogenies. In: Chung, K.M., Sasaki, Y. (eds.) ASIACRYPT 2024, Part III. LNCS, vol. 15486, pp. 272–303. Springer, Singapore (Dec 2024). https://doi.org/10.1007/978-981-96-0891-1_9
26. 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
27. Panny, L., Petit, C., Stopar, M.: KLaPoTi: An asymptotically efficient isogeny group action from 2-dimensional isogenies. CiC **2**(3), 5 (2025). https://doi.org/10.62056/ahp2wakrz
28. Wesolowski, B.: The supersingular isogeny problem in time and memory $p^{1/3+o(1)}$. Cryptology ePrint Archive, Paper 2026/1486 (2026), https://eprint.iacr.org/2026/1486

A Omitted Proofs

A.1 Characterising diagonal isogenies at the quaternion level

In this subsection, we work to demonstrate Lemma 1. Let (P, Q) be a basis of $E_0[2^{e+2}]$ such that $[2^{e+1}]Q = (0 : 0 : 1)$. We denote as $\iota : (x, y) \mapsto (-x, \sqrt{-1}y)$ and $\pi : (x, y) \mapsto (x^p, y^p)$ the endomorphisms of E_0 corresponding to $\mathbf{i}$ and $\mathbf{j}$. If the first isogeny of Φ is not a gluing, then it is one of the following:

– *A diagonal isogeny*: An isogeny between product of elliptic curves $\mathrm{Diag}(\psi_1, \psi_2) : E_0^2 \to E \times E'$ of the form

$$\mathrm{Diag}(\psi_1, \psi_2) = \begin{pmatrix} \psi_1 & 0 \\ 0 & \psi_2 \end{pmatrix}$$

$$\ker\left(\mathrm{Diag}(\psi_1, \psi_2)\right) = \left\langle \big(\ker(\psi_1), 0\big), \big(0, \ker(\psi_2)\big) \right\rangle$$

– *The canonical 2-endomorphism over products* : Defined over all products, $\Delta : E_0^2 \to E_0^2$ is induced by Kani's Lemma on a commutative diagram full of identity.

$$\begin{array}{ccc} E & \xrightarrow{[1]} & E \\ {\scriptstyle [1]}\downarrow & & \downarrow{\scriptstyle [1]} \\ E & \xrightarrow[[1]]{} & E \end{array}$$

$$\Delta(X, Y) = (X + Y, X - Y)$$

$$\Delta = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$\text{with } \ker(\Delta) = [2^{e+1}]\big\langle (P,P),(Q,Q)\big\rangle$$

– *The twisted* 2*-endomorphism over* E_0 : Specific to E_0 and consequence of $|\mathrm{Aut}(E_0)| = 4$ unlike almost all other curves, $\nabla : E_0^2 \to E_0^2$ is induced by Kani's Lemma on the following commutative diagram.

$$\begin{array}{ccc} E_0 & \xrightarrow{[1]} & E_0 \\ {\scriptstyle\iota}\downarrow & & \downarrow{\scriptstyle\iota} \\ E_0 & \xrightarrow[[1]]{} & E_0 \end{array}$$

$$\nabla(X,Y) = \big(X + \iota(Y), \iota(Y) - X\big)$$

$$\nabla = \begin{pmatrix} 1 & \iota \\ -\iota & -1 \end{pmatrix}$$

$$\text{with } \ker(\nabla) = [2^{e+1}]\big\langle (Q,\iota(Q)),(P,\iota(P))\big\rangle = [2^{e+1}]\big\langle (Q,Q),(P,P+Q)\big\rangle.$$

Proposition 10. *The first isogeny of the* $(2,2)$*-chain is not a gluing if and only if* $\theta([2^{e+1}]Q) = [2^{e+1}]Q$.

Proof. Looking at all possible $\ker(\Phi)[2]$ with basis $\mathcal{B}$. Let $\theta \begin{pmatrix} P \\ Q \end{pmatrix} = \begin{pmatrix} \theta_{1,1} & \theta_{1,2} \\ \theta_{2,1} & \theta_{2,2} \end{pmatrix} \begin{pmatrix} P \\ Q \end{pmatrix}$.

Looking at $u, \theta_{1,1}, \theta_{1,2}, \theta_{2,1}, \theta_{2,2} \mod 2$, we must have that $u = \theta_{1,1}\theta_{2,2} - \theta_{2,1}\theta_{1,2} = 1 \mod 2$. All possible values are listed in Table 6, which finishes the proof.

□

$(u, \theta_{1,1}, \theta_{1,2}, \theta_{2,1}, \theta_{2,2})$	$\ker(\psi)[2]$	ψ
$(1,1,0,0,1)$	$[2^{e+1}]\langle (P,P),(Q,Q)\rangle$	Δ
$(1,1,0,1,1)$	$[2^{e+1}]\langle (P,P),(Q,P+Q)\rangle$	glue
$(1,1,1,0,1)$	$[2^{e+1}]\langle (P,P+Q),(Q,Q)\rangle$	∇
$(1,0,1,1,0)$	$[2^{e+1}]\langle (P,Q),(Q,P)\rangle$	glue
$(1,0,1,1,1)$	$[2^{e+1}]\langle (P,Q),(Q,P+Q)\rangle$	glue
$(1,1,1,1,0)$	$[2^{e+1}]\langle (P,P+Q),(Q,P)\rangle$	glue

Table 6. First isogeny of the dim 2 chain, depending on the nature of $\theta \mod 2$.

We now desire to express the geometric condition of Proposition 10 as a quaternionic condition. To do so, we will exploit the Deuring correspondence once more.

Proposition 11. *Let $(P', Q') = [2^e](P, Q)$ be a basis of $E_0[4]$. π and ι behave on this basis as follows.*

1. $\iota \begin{pmatrix} P' \\ Q' \end{pmatrix} = \begin{pmatrix} \iota_{1,1} & \iota_{1,2} \\ \iota_{2,1} & -\iota_{1,1} \end{pmatrix} \begin{pmatrix} P' \\ Q' \end{pmatrix}$ *with* $\iota_{1,1}, \iota_{1,2}$ *odd and* $\iota_{2,1} = 2 \mod 4$.
2. $\pi \begin{pmatrix} P' \\ Q' \end{pmatrix} = \begin{pmatrix} \pi_{1,1} & \pi_{1,2} \\ \pi_{2,1} & -\pi_{1,1} \end{pmatrix} \begin{pmatrix} P' \\ Q' \end{pmatrix}$ *with* $\pi_{1,1}, \pi_{1,2}$ *odd and* $\pi_{2,1} = 0 \mod 4$.

Proof. 1. Looking at $\iota(E_0[2])$, we see that $[2^{e+1}]\iota(P) = [2^{e+1}](P + Q)$ while $[2^{e+1}]\iota(Q) = [2^{e+1}]Q$, which gives us $\iota_{1,1}, \iota_{1,2}, \iota_{2,1} \mod 2$. Additionally, as $\iota^2 = [-1]$, which implies that $\iota_{1,1}^2 + \iota_{2,1}\iota_{1,2} = -1 \mod 2^e$, which seen $\mod 4$ implies that $1 + \iota_{2,1}\iota_{1,2} = -1 \mod 4$, meaning that $\iota_{2,1} = 2 \mod 4$. This also implies that $(\iota_{1,1} + \iota_{2,2}) = 0 \mod 2^e$.

2. Looking at $\pi(E_0[2])$, we see that $[2^{e+1}]\pi(P) = [2^{e+1}](P+Q)$ while $\pi([2^{e+1}]Q) = [2^{e+1}]Q$, which gives us the values of $\pi_{1,1}, \pi_{1,2}, \pi_{2,1} \mod 2$. Also, $\pi^2 = [-p]$, which implies $\beta(\pi_{1,1} + \pi_{2,2}) = 0 \mod 2^e$. Additionally, we have that $\pi_{1,1}^2 + \pi_{1,2}\pi_{2,1} = -p \mod 2^e$, which considered $\mod 4$ and using $p = 3 \mod 4$ implies that $1 + \pi_{1,2}\pi_{2,1} = 1 \mod 4$, meaning that $\pi_{1,2}\pi_{2,1} = 0 \mod 4$. As $\pi_{1,2}$ is odd, then $\pi_{2,1} = 0 \mod 4$.

Corollary 2. *The evaluation of the canonical basis of $\mathcal{O}_0$ on $E_0[2]$ is as follows. (For readably, we omit the 2^{e+1} scaling factor.)*

$$\mathsf{id}(P) = P, \quad \mathsf{id}(Q) = Q,$$

$$\iota(P) = P + Q, \quad \iota(Q) = Q,$$

$$\left(\frac{\iota + \pi}{2}\right)(P) = \left[\frac{\pi_{1,1} + \iota_{1,1}}{2}\right] P + \left[\frac{\pi_{1,2} + \iota_{1,2}}{2}\right] Q,$$

$$\left(\frac{\iota + \pi}{2}\right)(Q) = P + \left[\frac{-(\iota_{1,1} + \pi_{1,1})}{2}\right] Q,$$

$$\left(\frac{1 + \iota \circ \pi}{2}\right)(P) = \left[\frac{\pi_{1,1}\iota_{1,1} + 1}{2}\right] P + \left[\frac{\pi_{1,2}\iota_{1,1} - \pi_{1,1}\iota_{1,2}}{2}\right] Q,$$

$$\left(\frac{1 + \iota \circ \pi}{2}\right)(Q) = P + \left[1 + \frac{\pi_{1,1}\iota_{1,1} + 1}{2}\right] Q.$$

The fact that θ is an endomorphism of odd degree imposes some conditions on its representation in the canonical basis of $\mathcal{O}_0$.

Proposition 12. *An odd norm endomorphism $\theta \in \mathcal{O}_0$ cannot be of either of the following forms:*

- $x_\theta = y_\theta = 0 \mod 2$.
- $z_\theta = w_\theta = 1 \mod 2$.
- $x_\theta = y_\theta = 1 \mod 2$ *and* $z_\theta = w_\theta = 0 \mod 2$.

Proof. By definition of the reduced norm in $\mathcal{B}_{p,\infty}$, we have that

$$n(\theta) = x_\theta(x_\theta + w_\theta) + y_\theta(y_\theta + z_\theta) + \frac{p+1}{4}(z_\theta^2 + w_\theta^2)$$

Reducing modulo 2, we obtain:

$$x_\theta(x_\theta + w_\theta) + y_\theta(y_\theta + z_\theta) = 1 \mod 2$$

and we see that all cases we denoted above do not satisfy this equation. □

Under the assumption that θ is of odd norm, there is therefore a simple characterisation of Proposition 10 on the quaternion side.

Proposition 13.

$$\theta([2^{e+1}]Q) = [2^{e+1}]Q \Longleftrightarrow z_\theta = w_\theta = 0 \mod 2 \Longleftrightarrow \theta \in \mathbb{Z}[\mathbf{i}, \mathbf{j}]$$

Proof. A direct consequence of Corollary 2 when $(z_\theta, w_\theta) \neq (1, 1) \mod 2$. □

Combining Proposition 13 together with Proposition 11 gives us a proof of Lemma 1

A.2 Omitted proofs from Section 3

Lemma 3. *Let $\alpha = a_\alpha + b_\alpha \mathbf{i} + c_\alpha \mathbf{j} + d_\alpha \mathbf{ij} = x_\alpha + y_\alpha \mathbf{i} + z_\alpha \frac{\mathbf{i}+\mathbf{j}}{2} + w_\alpha \frac{1+\mathbf{ij}}{2} \in \mathcal{O}_0$, with $n(\alpha) = Nr$. Let (s, t) be solution of the equation $2a_\alpha s + 2b_\alpha t = 2^e - 2r \mod N$, with $kN = 2^e - 2r - 2a_\alpha s - 2b_\alpha t$ and $z = 2k - s^2 - t^2$. Finally, let s_c, t_c be such that $s_c^2 + t_c^2 = z$.*

Then $(s + \mathbf{i}t) \cdot \overline{\alpha} \in \mathcal{O}_0 \backslash \mathbb{Z}[\mathbf{i}, \mathbf{j}]$ and $s_c = s \mod 2$ and $t_c = t \mod 2$ (up to permutation of s_c, t_c) [9] if and only if one of the following two conditions holds:

- *$z_\alpha + w_\alpha$, $s + t$ and k odd and $z = 1 \mod 4$.*
- *N, z_α, w_α and k are odd, $s + t = 1 \mod 2$ and $z = 1 \mod 4$.*

Proof (Lemma 3).

For this proof, we consider the following set of equations (from the definitions of k, z):

$$-2a_\alpha s - 2b_\alpha t = 2r + kN \mod 4 \tag{13}$$

$$2k - s^2 - t^2 = z \mod 4 \tag{14}$$

Furthermore, note that the shared parity conditions on s_c, s and t_c, t can be rewritten in terms of (z, s, t) only as follows:

$$\begin{cases} z = 0 \mod 4 \text{ and } (s, t) = (0, 0) \mod 2 \\ z = 1 \mod 4 \text{ and } s + t = 1 \mod 2 \\ z = 2 \mod 4 \text{ and } (s, t) = (1, 1) \mod 2 \end{cases} \tag{15}$$

Looking at all the possible configurations, we have that:

[9] This second condition is required to ensure the existence of a solution of the Qlapoti norm equation, as $a_{\gamma_1} = (s_c + s)/2$ and $b_{\gamma_1} = (t_c + t)/2$.

– **Case** $(z_\alpha, w_\alpha) = (0,0) \mod 2$: Then $\alpha \in \mathbb{Z}[\mathbf{i},\mathbf{j}]$ and hence $(s+\mathbf{i}t)\cdot\overline{\alpha} \in \mathbb{Z}[\mathbf{i},\mathbf{j}]$. It therefore does not satisfy the first condition of Lemma 3.
– **Case** $N = 0 \mod 2$ and $(z_\alpha, w_\alpha) = (1,1) \mod 2$: This also does not satisfy the first condition of Lemma 3. Indeed, as $\alpha \in O_0$, we have $2a_\alpha$, $2d_\alpha$ and w_α of the same parity and $2b_\alpha$, $2c_\alpha$, and z_α of the same parity, so that Eq. (13) shows that $s + t = 0 \mod 2$. Therefore, $s = t \mod 2$, implying:

$$(c_\alpha + \mathbf{i}d_\alpha)\cdot(s+\mathbf{i}t) = sc_\alpha - td_\alpha + (sd_\alpha + tc_\alpha)\mathbf{i} \in \mathbb{Z}[\mathbf{i}]$$

and thus, since $a_\alpha = d_\alpha \mod 2$ and $b_\alpha = c_\alpha \mod 2$, that $(s+\mathbf{i}t)\cdot\overline{\alpha} \in \mathbb{Z}[\mathbf{i},\mathbf{j}]$.
– **Case** $N = 0 \mod 2$ and $z_\alpha + w_\alpha = 1 \mod 2$: This implies N even and $2a_\alpha + 2b_\alpha = 1 \mod 2$ thus, by Eq. (13) either s or t is even. Furthermore, s and t cannot be simultaneously even, because of the first condition of Lemma 3, so we have that $s + t = 1 \mod 2$. This implies that $z = 1 \mod 4$ (provided $k = 1 \mod 2$, but the other case is absurd since $z = 3 \mod 4$ is not a sum of 2 squares). $s + t = 1 \mod 2$ and $z = 1 \mod 4$ is in Eq. (15) . Thus this case satisfies all conditions of Lemma 3.
– **Case** $N = 1 \mod 2$ and $(z_\alpha, w_\alpha) = (1,1) \mod 2$: In the case where $r = 1 \mod 2$, then $n(\alpha) = r \cdot N$ is odd, but in this case, it is impossible for $(z_\alpha, w_\alpha) = (1,1) \mod 2$ following Proposition 12. Hence, such case never occurs. In the case there $r = 0 \mod 2$, consider Eq. (13):
  • if $k = 0 \mod 2$: this implies $s = t \mod 2$, which, by the same reasoning as when $N = 0 \mod 2$ and $(z_\alpha, w_\alpha) = (1,1) \mod 2$, implies that $(s + t\mathbf{i})\cdot\overline{\alpha} \in \mathbb{Z}[\mathbf{i},\mathbf{j}]$ and hence not satisfying the first condition of Lemma 3.
  • if $k = 1 \mod 2$: We have that $s + t = 1 \mod 2$. But then, modulo 4, $z = 2k - s^2 - t^2 = 2 - 1 = 1 \mod 4$ (case 2 of Eq. (15)). Since $2a_\alpha, 2b_\alpha, 2c_\alpha$ and $2d_\alpha$ are odd (as z_α and w_α are), $s + t = 1 \mod 2$ implies that $(s+t\mathbf{i})\overline{\alpha} \notin \mathbb{Z}[\mathbf{i},\mathbf{j}]$. This case therefore satisfies all conditions of Lemma 3.
– **Case** $N = 1 \mod 2$ and $z_\alpha + w_\alpha = 1 \mod 2$: Without loss of generality, assume $w_\alpha = 0 \mod 2$. Using Eq. (13), we get that $t = k \mod 2$, which implies that modulo 4, $z = 2k - s^2 - t^2 = 2k - s^2 - k^2 = k(2-k) - s^2 \mod 4$. Considering that only 0 and 1 are squares modulo 4, we get, still modulo 4 that

$$z = \begin{cases} -s^2 = 0,3 \mod 4 & \text{if } k = 0 \mod 2 \\ 1 - s^2 = 0,1 \mod 4 & \text{if } k = 1 \mod 2 \end{cases}$$

Thus the only case which satisfies the second condition of Lemma 3 is when $z = 1 \mod 4$ and $s \neq t \mod 2$. Indeed, the case $z = 3 \mod 4$ does not allow z to be a sum of two squares. The case $-s^2 = 0 \mod 4$ implies s, k and thus t are even, contradicting the first condition of Lemma 3. Similarly $1 - s^2 = 0 \mod 4$ implies s, k and thus t are odd, but this contradicts Eq. (15). Finally, if modulo 4 $z = 1 - s^2 = 1 \mod 4$ and $k = t$ odd, then s, t are of different parities, with the parity of s matching the one of z_α and $2b_\alpha$ and different from the one of w_α and a_α, thus implying that $(s + t\mathbf{i})(a_\alpha - b_\alpha\mathbf{i}) = (a_\alpha s - b_\alpha t) - (b_\alpha s + a_\alpha t)\mathbf{i} \notin \mathbb{Z}[\mathbf{i},\mathbf{j}]$, so that in this case, all conditions of Lemma 3 are satisfied.

□

Proposition 1. *Let $I_1 \sim I_2$ be left $\mathcal{O}_0$-ideal such that $n(I_1) + n(I_2) = 2^e$. Let $\mathcal{O}_0\theta = I_2\overline{I_1}$ with $\theta = a_\theta + b_\theta\mathbf{i} + c_\theta\mathbf{j} + d_\theta\mathbf{ij}$ primitive in $\mathcal{O}_0$. Then,*

$$\gcd(n(I_1), n(I_2)) = 1 \Longleftrightarrow \exists!\, x \in \{2a_\theta, 2b_\theta, 2c_\theta, 2d_\theta\} \text{ s.t. } x = 2 \mod 4$$

Proof (Proposition 1). As $n(I_1) + n(I_2) = 2^e$, they cannot share a commun odd divisor, so $n(I_1) = n(I_2) = 1 \mod 2$ is equivalent to their gcd being 1 and thus to $n(\theta) = 1 \mod 2$. As $\theta \in \mathcal{O}_0$ is primitive, there are two conditions:

- If both $(2a_\theta, 2d_\theta)$ and $(2b_\theta, 2c_\theta)$ are odd. Then, by the norm equation (see Proposition 12), $n(\theta)$ must be even.
- If either $(2a_\theta, 2d_\theta)$ or $(2b_\theta, 2c_\theta)$ (but not both) are odd, the other one can be set to even without loss of generality, so that:

$$\begin{aligned}
\gcd(n(I_1), n(I_2)) = 1 &\Longleftrightarrow n(\theta) = 1 \mod 2 \\
&\Longleftrightarrow a_\theta^2 + b_\theta^2 + p(c_\theta^2 + d_\theta^2) = 1 \mod 2 \\
&\Longleftrightarrow a_\theta^2 - d_\theta^2 = 1 \mod 2 \\
&\Longleftrightarrow a_\theta \neq d_\theta \mod 2 \\
&\Longleftrightarrow 2a_\theta \neq 2d_\theta \mod 4
\end{aligned}$$

which proves our point. □

A.3 Omitted proofs from Section 4

Lemma 8. *Let I an $\mathcal{O}_0$-ideal smallest in its equivalence class. Denote N its norm. Then its HNF (column-wise, upper triangular in basis $1, \mathbf{i}, \mathbf{j}, \mathbf{ij}$) is almost equal to a reduced basis, in the following sense: The first two HNF vectors have minimal norm, and the norms of the last 2 HNF vectors are at most $2N^2 < p/32$ larger than the third and forth minimum.*

More precisely, a reduced form is also almost an HNF, in the following sense:

1. *The reduced basis can be made upper triangular in basis $1, \mathbf{i}, \mathbf{j}, \mathbf{ij}$ by reordering if needed the first and if needed the last pair of vectors it contains, if they have equal norms.*
2. *The diagonal is equal up to sign, so flip signs of reduced basis vectors when needed. The basis is still reduced after these 2 steps.*
3. *In the resulting reduced form as in the HNF, the first two vectors are $N, N\mathbf{i}$.*
4. *The last two vectors of HNF and this reduced form have the same $\mathbf{j}$ and $\mathbf{ij}$ coordinates*
5. *The $1, \mathbf{i}$-coordinates of the last 2 vectors only differ by at most n (on each of these 4 coordinates).*

Proof. We start with proving the second part of the lemma.

The ideal I being minimal in its class, all its elements have norm at least N^2 since otherwise there would be a smaller equivalent ideal. Since N and $N\mathbf{i}$ are elements of norm N^2, they attain the first minimum. As they are linearly independent, they are (up to sign) the first two minima. They also are a diagonal basis of the subspace $\mathbb{Z}[\mathbf{i}] \cap I$ of I, so they are the first two HNF vectors too. This shows point 2. and parts of 1. and 0.

The next step requires to observe that for all elements in an $\mathcal{O}_0$-ideal, if the $\mathbf{j}$-coefficient of x is γ_x, then there exists an element y linearly independent of x whose coefficient $\mathbf{ij}$ is equal to γ. One can exchange the roles of $\mathbf{j}$ and $\mathbf{ij}$ in this.

Since I is smallest in its class, $4N^2 < p$. As the $1, \mathbf{i}$-coefficients can always be taken mod N in both the HNF and the reduced form since there is the vectors $N, N\mathbf{i}$ in the ideal, the norm of the last 2 minimal or HNF vectors is mostly determined by the coefficients of $\mathbf{j}$ and $\mathbf{ij}$.

In the HNF, the coefficient in $\mathbf{ij}$ of the 4th vector is minimal non-zero since gcd of the $\mathbf{ij}$-coefficients all vectors in the ideal. Denote this coefficient δ. Given the above observation, there is a linearly independent vector having δ as $\mathbf{j}$-coefficient. Since the third HNF vector has $\mathbf{j}$-coefficient equal to the gcd of all $\mathbf{j}$-coefficients of vectors linearly independent of the 4th vector at that state of the HNF computation (where the 4th vector already has $\mathbf{ij}$-coefficient δ), the $\mathbf{j}$-coefficient of the HNF is also δ and minimal among all $\mathbf{j}$-coefficients in the ideal (again up to sign of course). Furthermore, this implies that the $\mathbf{j}$-coordinate of the 4th vector at that state is divisible by δ and will therefore be set to 0 in the final HNF. Given there is a basis with $\mathbf{j}, \mathbf{ij}$-coefficients $\delta, 0$ and $0, \delta$ and the minimality of δ, the minimal basis also contains vectors with these coefficients to attain the third and fourth minima. This proves points 0. and 1. fully and also point 3.

Finally, the 4 remaining coefficients (the $1, \mathbf{i}$-coefficients of the last 2 vectors) are taken modulo N using $N, N\mathbf{i}$. For the reduced basis, they are taken minimal in absolute value, so in $[-N/2, N/2]$ while the HNF takes them in $[0, N[$. This proves point 4 and also implies a maximal difference in norm of $2N^2 < p/32$ between the norm of the last 2 HNF vectors and the last 2 lattice minima is at most $2N^2 < p/32$. □

Proposition 2. *Let $B_\alpha(N) \in \mathbb{N}$. Then, we can sample $\alpha_0 := \sum_{\kappa=1}^{4} c_\kappa \cdot \beta_\kappa = a_\alpha + \mathbf{i}b_\alpha + \alpha_2\mathbf{j} \notin \mathbb{Z}[\mathbf{i}, \mathbf{j}]$ such that $2(a_\alpha + b_\alpha) = 1 \mod 2$ and $2a_\alpha, 2b_\alpha \leq 2N$ and $n(\alpha_0) \leq 2N^2 + B_\alpha^2 p/4$.*

Proof (Proposition 2). We may take $c = (c_3, c_4) \in [0, B_\alpha/\sqrt{2}]^2$ such that the parity of c_3 and c_4 is not the same, we then get that $z_c\beta^I$ follows the first condition by Lemma 6. After choosing c_1, c_2 such that α_0 is equal to $\alpha_0(c) = z_c\beta^I \mod N\mathbb{Z}[\mathbf{i}]$, the first condition remains unchanged. We clearly get the second condition, and then we have that

$$n(\alpha_0) \leq 2N^2 + (c_3^2 + c_4^2)n(\mathbf{j}/2) \leq 2N^2 + B_\alpha^2 p/4$$

which proves the result. □

Proposition 6. *Let us take an odd $n_c^2 \in S_{\mathbf{i}}(a,b)$ for some $1 < a < b$, with n_c^2 dividing ℓ^I. Let $z_c \in \mathbb{Z}[\mathbf{i}]$ be an element of norm n_c and such that ℓ^I is the biggest ℓ such that $z_c\omega^I \in \ell\mathbb{Z}[\mathbf{i}]$.*[10] *Then, $\lambda_1^I(c) = \lambda_1^I/n_c$.*

Proof (Proposition 6).

We will show that the vector $z_c\omega^I/n_c^2$, which is of norm λ_1^I/n_c is the minima of the lattice $\Psi(\Lambda^I(c))$. First since n_c^2 divides ℓ^I, this element is indeed in $\mathbb{Z}[\mathbf{i}]$. Then, $z_c\omega^I/n_c^2 \in \Psi(\Lambda^I(c))$ as $\mathrm{tr}(\overline{z_c\omega^I}z_c\beta^I/n_c^2) = (n_c^2/n_c^2)\mathrm{tr}(\omega^I\beta^I) = 0 \mod 4N$. Furthermore, it is primitive in the lattice. Indeed, let ℓ be maximal such that

1. $z_c\omega^I \in n_c^2\ell\mathbb{Z}[\mathbf{i}]$
2. $\mathrm{tr}(\omega^I\beta^I) = 0 \mod 4N\ell$

Let us show that $\ell = 1$. By our choice of z_c, we have that $\ell|\ell^I$ and thus that $\omega^I \in \ell\mathbb{Z}[\mathbf{i}]$. Furthermore,

$$\mathrm{tr}((\omega^I/\ell')z^I) = 0 \mod 4N$$

meaning that the point ω^I/ℓ is inside $\Psi(\Lambda^I)$, breaking the minimality of ω^I.

We therefore have that $z_c\omega^I/n_c^2$ is primitive in our lattice.

Finally, it must be the smallest element as if it were not, it would be bigger than the second minimum of the lattice $\Psi(\Lambda^I(c))$ which is bigger than $\sqrt{8N/\pi}$ by Minkowski's second theorem. However, the same theorem also gives that $\lambda_1^I \leq 4\sqrt{N/\pi}$. Thus, since $n_c^2 > 2$, we have that $\lambda_1^I/n_c \leq \sqrt{8N/\pi}$ which is a contradiction.

Proposition 7. *Let us take an odd $n_c^2 \in S_{\mathbf{i}}(a,b)$ for some $1 < a < b$, such that n_c^2 is coprime to ℓ^I. Let $z_c \in \mathbb{Z}[\mathbf{i}]$ be an element of norm n_c and such that ℓ^I is the biggest ℓ such that $z_c\omega^I \in \ell\mathbb{Z}[\mathbf{i}]$.*[5] *Then, $z_c\omega^I$ is a primitive element of $\Psi(\Lambda^I(c))$.*

Proof (Proposition 7). By Proposition 3, it is clear that $z_c\omega^I$ is an element of $\Psi(\Lambda^I(c))$. Assume that it is not primitive and that there exists $\ell > 1$ such that:

1. $z_c\omega^I \in \ell\mathbb{Z}[\mathbf{i}]$
2. $n_c^2\mathrm{tr}(\overline{\omega^I}\beta^I) = 0 \mod 4N\ell$

Since ℓ and n_c^2 are coprime, there exists N_1, N_2 such that:

1. $4N = N_1N_2$
2. $\gcd(N_1, N_2) = 1$
3. $\gcd(N_1\ell, n_c^2) = 1$
4. $\gcd(N_2, \ell) = 1$

By our choice of z_c, we have that $\ell|\ell^I$ and thus that $\omega^I \in \ell\mathbb{Z}[\mathbf{i}]$. Thus, we get $n_c^2\mathrm{tr}(\overline{\omega^I}\beta^I) = 0 \mod N_1\ell$ which implies that $\mathrm{tr}(\overline{\omega^I}\beta^I/\ell) = 0 \mod N_1$ since n_c^2 is invertible mod $N_1\ell$. Then, since ℓ is invertible $\mod N_2$ we get that $\mathrm{tr}(\overline{\omega^I}\beta^I) = \ell\mathrm{tr}(\overline{\omega^I}\beta^I/\ell) = 0 \mod N_2$ which implies that $\mathrm{tr}(\overline{\omega^I}\beta^I/\ell) = 0 \mod N_2$.

Thus, by the CRT, we have that $\mathrm{tr}(\overline{\omega^I}\beta^I/\ell) = 0 \mod N_1N_2$ which would mean that ω^I/ℓ is an element of $\Psi(\Lambda^I)$ which contradicts the minimality of ω^I.

[10] As enabled by Proposition 5

A.4 Detailed heuristic estimation of the bound of $\mathbf{P}_{\text{fail-ideal}}$.

For any integer M, the expected number of squareroots for a random $n_0 \in \mathbb{Z}/M\mathbb{Z}$ is 1. Thus, the expected value of $\sum_{n_0 \in S} \#\{\lambda \in \mathbb{Z}/M\mathbb{Z} \mid \lambda^2 = n_0\}$ when S is a uniformly random subset of $\mathbb{Z}/M\mathbb{Z}$ is $\#S$. And so assuming that the set of n_0 appearing in the sum from Proposition 9 behave like a random set, we expect that

$$\sum_{p^{1/4} \leq N \leq \frac{\sqrt{p}}{2}} \sum_{\substack{g \mid 4N \\ g \leq B(p,N)}} \frac{g^2}{2} \sum_{s_0^2 + t_0^2 \leq B(p,N)^2} \#\{\lambda \in \mathbb{Z}/\frac{4N}{g}\mathbb{Z}^\times \mid \lambda^2 = -\frac{s_0^2 + t_0^2}{p}\}$$

$$\leq \sum_{p^{1/4} \leq N \leq \frac{\sqrt{p}}{2}} \sum_{\substack{g \mid 4N \\ g \leq B(p,N)}} \frac{g^2}{2} \#\{s_0, t_0 \in \mathbb{Z} | s_0^2 + t_0^2 \leq B(p,N)^2\}$$

Then, with $\#\{s_0, t_0 \in \mathbb{Z} | s_0^2 + t_0^2 \leq B(p,N)^2\} \leq 2B(p,N)^2 + 1$, we get the upperbound

$$\sum_{p^{1/4} \leq N \leq \frac{\sqrt{p}}{2}} B(p,N)^2 (B(p,N)^2 + \frac{1}{2})$$

For cryptographic sizes of p chosen such that f is as small as possible (which is the case in SQIsign), we can reasonnably assume that $\log^2 N \ll B_\alpha(N) \sim \sqrt{N}$ when $N > p^{1/4}$, and so we can safely assume that $B(p,N) \leq 2B_1(N)/B_\alpha(N)$. Then, with $B_1(N) = \sqrt{N/2}$, we get $B(p,N) \leq \sqrt{2}\frac{\sqrt{N}}{B_\alpha(N)}$. And finally, using $B_\alpha(N) = \sqrt{(2^{e+3}N - 16N^2)/p}$ and $N < \sqrt{p}/2$ which implies $16N^2 \ll 2^{e+3}N$, we can get $B_\alpha(N) > \sqrt{\frac{N}{f}}$, which leads to

$$B(p,N) \leq \sqrt{2f}$$

Importantly, this bound is independant of N, and so we get that the number of failing ideal class can be upper bounded by

$$\frac{12}{p}\left(\frac{1}{4}\sqrt{p}(\log p + 8) + \frac{1}{4}h(-p)f\log(\sqrt{2f}) + \sqrt{p}f(2f + \frac{1}{2})\right)$$

that we can rewrite as

$$\frac{1}{\sqrt{p}}\left(3(\log p + 8) + 12f(2f + \frac{1}{2}) + 3f\log(4\sqrt{f})\frac{h(-p)}{\sqrt{p}}\right).$$

From Dirichlet's class number formula and the explicit constants from [17] we can derive the bound

$$h(-p) \leq \frac{1}{2\pi}\sqrt{p}\log p,$$

that will be true for the primes p we consider for cryptography.

This leads to our final estimate Eq. (8).

A.5 Distribution of the norm of (s, t) with respect to N

Let N, a, b be integers such that $\gcd(a, b, 4N) = 1$. Let Λ be the rank 2 lattice describing the solution of the equation $ax+by = 0 \mod 4N$. We ask for $\lambda_1(\Lambda) \geq \sqrt{N/2}$, meaning that Λ is balanced. For every $c \in \mathbb{Z}_{4N}$ odd, let (s, t) is a solution of small norm to the linear equation $as+bt = c \mod 4N$ computed using Babai's algorithm on a Minkowski-reduced basis and let $\epsilon = \sqrt{s^2 + t^2/N}$.

We are interested in studying $F_\epsilon(t) = \Pr[\epsilon \leq t]$. To do so, we will study its pdf, f_ϵ. As illustrate by Fig. 1 its behaviour is closely linked to the minima of Λ. We can split it into 3 part:

- For $t \leq \lambda_1(\Lambda)/\sqrt{4N}$, we see that then f_ϵ behaves linearly. More specifically, our experimental analysis seams to suggest that over this regime, we have that
$$f_\epsilon(t) \simeq \frac{\pi}{2} \cdot t$$
Hence, by the limit we set on Λ, we have that for $t \leq 1/\sqrt{8}$, we have that $F_\epsilon(t) \simeq \frac{\pi}{4} t^2$. In Section 3.3, all our evaluation fall into this case.
- For $t \in [\lambda_1(\Lambda)/\sqrt{4N}, \lambda_2(\Lambda)/\sqrt{4N}]$, the function decrease towards a constant probability.
- For $t \geq \lambda_2(\Lambda)/\sqrt{4N}$, the function sharply decrease towards zero.

B A failed resampling mechanism for Qlapoty

In this section, we present a mechanism that, given α_0, resamples α as an element of the form $\alpha := \alpha_0 + N(c_1 + \mathbf{i}c_2)$. We are thus searching for elements $\beta_i := (a_i + b_i\mathbf{i})N + \alpha$ that satisfy the norm equation $n(\beta_1) + n(\beta_2) = 2^e N$.

This mechanism appears promising because $\Lambda(\alpha) = \Lambda(\alpha_0)$, even though its target is changed. Furthermore, note that $n(\alpha) \leq n(\alpha_0) + N^2(c_1^2 + c_2^2)$, which provides significantly more flexibility for resampling. Nevertheless, this resampling method is unsuited for solving the Qlapoty norm equation, as we will demonstrate.

Let α_0 be a generator of I and $\alpha := \alpha_0 + N(c_1 + \mathbf{i}c_2)$. Let (s_0, t_0) be the smallest solution to

$$2a_{\alpha_0}s + 2b_{\alpha_0}t \equiv 2^e - 2n(\alpha_0)/N \pmod N,$$

and let (s, t) be the analogous solution for α. Then, we have

$$\begin{aligned} 2^e - 2n(\alpha)/N &\equiv 2^e - 2n(\alpha_0)/N - 4a_{\alpha_0}c_1 - 4b_{\alpha_0}c_2 \pmod N \\ &\equiv 2a_{\alpha_0}s_0 + 2b_{\alpha_0}t_0 - 4a_{\alpha_0}c_1 - 4b_{\alpha_0}c_2 \pmod N \\ &\equiv 2a_{\alpha_0}(s_0 - 2c_1) + 2b_{\alpha_0}(t_0 - 2c_2) \pmod N. \end{aligned}$$

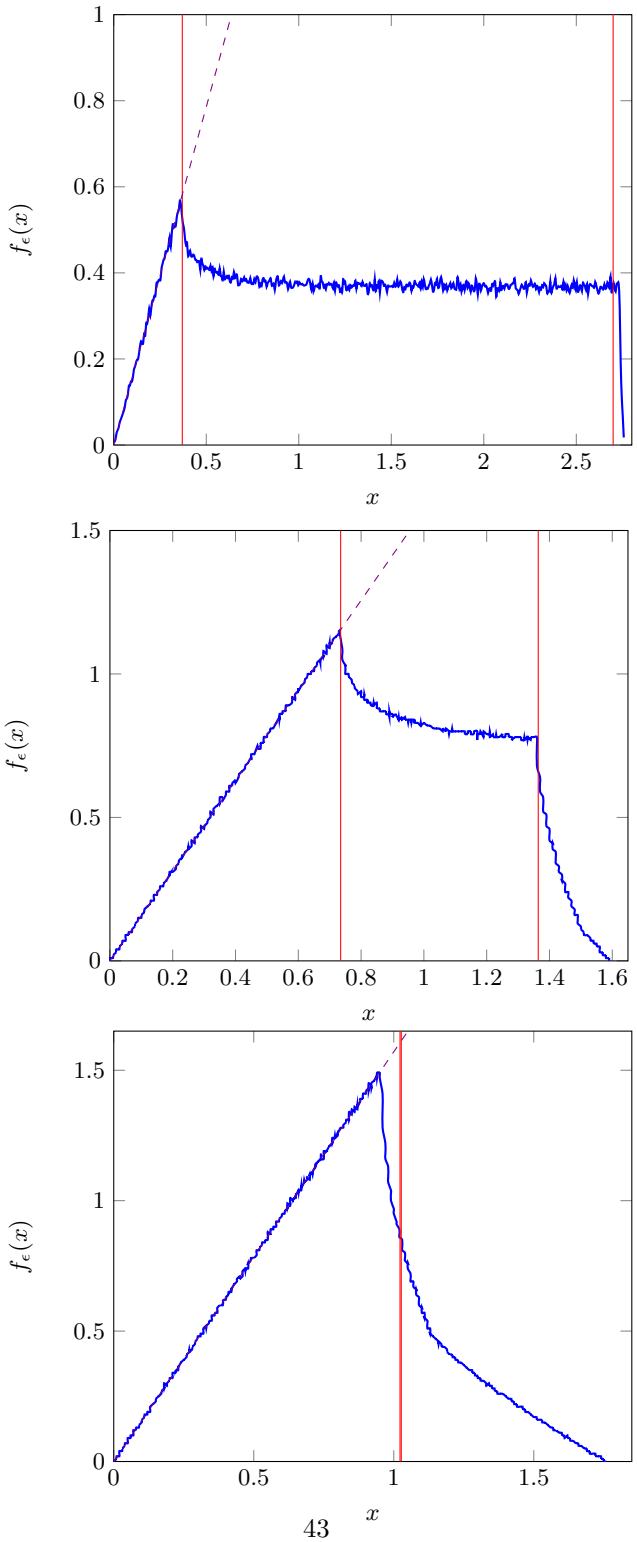


Fig. 1. Distribution $f_\epsilon(x)$ over lattices Λ with $\lambda_1(\Lambda) \in [\sqrt{N/2}, \sqrt{8N}]$. The red lines denotes the values $\lambda_i(\Lambda)/\sqrt{4N}$, and the purple line represents $\frac{\pi}{2}x$. All distributions were computed using 10^7 points.

It is therefore highly likely that, for $c_1, c_2 \ll \sqrt{N}$, this yields the minimal solution, implying that $s = s_0 - 2c_1$ and $t = t_0 - 2c_2$[11]. We then obtain

$$\begin{aligned}
Nk &= 2^e - 2n(\alpha)/N - 2a_\alpha s - 2b_\alpha t \\
&= 2^e - 2n(\alpha_0)/N - 2N(c_1^2 + c_2^2) - 4a_{\alpha_0} c_1 - 4b_{\alpha_0} c_2 \\
&\quad - 2a_{\alpha_0} s - 2Nc_1 s - 2b_{\alpha_0} t - 2Nc_2 t \\
&= 2^e - 2n(\alpha_0)/N - 2a_{\alpha_0}(s + 2c_1) - 2b_{\alpha_0}(t + 2c_2) \\
&\quad - 2N(c_1^2 + c_2^2) - 2Nc_1 s - 2Nc_2 t \\
&= 2^e - 2n(\alpha_0)/N - 2a_{\alpha_0} s_0 - 2b_{\alpha_0} t_0 - 2N(c_1^2 + c_2^2) - 2Nc_1 s - 2Nc_2 t.
\end{aligned}$$

Therefore, the input to Cornacchia's algorithm will be

$$\begin{aligned}
2k - s^2 - t^2 &= \frac{1}{N}\left(2^{e+1} - 4n(\alpha_0)/N - 4a_{\alpha_0} s_0 - 4b_{\alpha_0} t_0\right) \\
&\qquad - 4(c_1^2 + c_2^2) - 4c_1 s - 4c_2 t - s^2 - t^2 \\
&= 2k_0 - (2c_1 + s)^2 - (2c_2 + t)^2 \\
&= 2k_0 - s_0^2 - t_0^2,
\end{aligned}$$

which is constant. Hence, this resampling mechanism fixes z, making it unpractical for Qlapoty.

C Compiled Heuristics

In this section, we collect and summarize the heuristic assumptions used throughout our analysis of Qlapoty:

1. We rely on Assumption 1, which we verified experimentally with $C = 1$ for random values of N. For most values of N, this assumption would likely hold if the interval length were linear in $\log N$, though it might not be sufficient in the worst-case scenario (where N is the product of the smallest primes congruent to 1 (mod 4)).
2. We assume a specific distribution for f_ϵ when ϵ is small, as detailed in Appendix A.5, which has also been verified experimentally. A similar assumption and analysis for f_δ are used in [7, Section 4].
3. We assume that the distribution of our sampling function $\lambda \mapsto \lambda^{-1}(2^e \pm N) - 2\lambda r_0 \pmod{4N}$ is close to uniform over the appropriate subset of $4N$. This assumption is likewise supported by experimental verification.
4. We assume that the input to Cornacchia's algorithm, $z = 2k - s^2 - t^2$, behaves like a random integer equal to 1 mod 4. The number of attempts required by Cornacchia's algorithm supports this heuristic.
5. To obtain Eq. (8), We assume that the set of n_0 appearing in the sum from Proposition 9 behave like a random set. Indeed, as for any integer M, the expected number of square roots for a random $n_0 \in \mathbb{Z}/M\mathbb{Z}$ is 1, The expected value of $\sum_{n_0 \in S} \#\{\lambda \in \mathbb{Z}/M\mathbb{Z} \mid \lambda^2 = n_0\}$ when S is a uniformly random subset of $\mathbb{Z}/M\mathbb{Z}$ is $\#S$.

[11] This result was furthermore verified experimentally.

D Omitted algorithms

For completeness, let us first recall in Algorithm 5 pseudocode for the original Qlapoti norm equation solver as described and analysed (but not implemented) by [7]. Then, we give the pseudocode for some subalgorithms which are useful to compute Qlapoty efficiently.

Algorithm 5 Qlapoti (J) [7, Algorithm 2]

Input: Left O_0-ideal J.
Output: Equivalent ideals I_1 and I_2 with $n(I_1) + n(I_2) = 2^e$.

1: $I \leftarrow \mathsf{SmallestEquivIdeal}(J)$; ▷ Lemma 11,12 [10]
2: $N \leftarrow n(I)$;
3: $\alpha \leftarrow \mathsf{SmallGenerator}(I)$; ▷ write $\alpha = a_\alpha + b_\alpha \mathbf{i} + c_\alpha \mathbf{j} + d_\alpha \mathbf{ij}$ for $a_\alpha, b_\alpha, c_\alpha, d_\alpha \in \frac{1}{2}\mathbb{Z}$
4: $r \leftarrow n(\alpha)/N$;
5: **if** $\gcd(2a_\alpha, N) \neq 1$ and $\gcd(2b_\alpha, N) \neq 1$ **goto** line 3;
6: Compute shortest (s, t) such that $2a_\alpha s + 2b_\alpha t = 2^e - 2r \pmod N$ ▷ See [7, Lemma 2]
7: Set $z := 2(2^e - 2r - 2a_\alpha s - 2b_\alpha t)/N - s^2 - t^2$;
8: **if** $z < 0$ **goto** line 3;
9: **if** $z \equiv 0 \pmod 4$ and not $s = t = 0 \pmod 2$ **goto** line 3
10: **if** $z \equiv 1 \pmod 4$ and not $s \neq t \bmod 2$ **goto** line 3
11: **if** $z \equiv 2 \pmod 4$ and not $s = t = 1 \pmod 2$ **goto** line 3
12: **if** $z \equiv 3 \pmod 4$ **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: **end if**
16: $z_0, z_1 \leftarrow \mathsf{sol}$ ▷ $z_0^2 + z_1^2 = z$
17: **if** $z_0 \not\equiv s \pmod 2$ **then** swap z_0, z_1
18: **end if**
19: $a_1 \leftarrow (z_0 + s)/2$, $b_1 \leftarrow (z_1 + t)/2$;
20: $a_2 \leftarrow s - a_1$, $b_2 \leftarrow t - b_1$
21: $\beta_1 = n(a_1 + b_1 i) + \alpha$, $\beta_2 = n(a_2 + b_2 i) + \alpha$;
22: $I_1 = \chi_I(\beta_1)$, $I_2 = \chi_I(\beta_2)$
23: **return** I_1, I_2;

Algorithm 6 ComputeLattice

Require: $\alpha \in I$ such that $\gcd(\mathrm{tr}(\alpha), \mathrm{tr}(\mathbf{i}\alpha), 2N) = 1$.

Ensure: Λ the lattice corresponding to $\sigma \mapsto \mathrm{tr}(\overline{\sigma}\alpha_0) = 0 \mod 4N$ together with (ρ_1, ρ_2), solutions such that $\mathrm{tr}(\overline{(\rho_1 + \mathbf{i}\rho_2)}\alpha_0) = 1$

1: $g_a \leftarrow \gcd(\mathrm{tr}(\alpha), 4N)$, $a' \leftarrow \mathrm{tr}(\alpha)/g_a$

2: $c \leftarrow -\mathrm{tr}(\mathbf{i}\alpha) \cdot (a'^{-1} \mod (4N/g_a)) \mod 4N/g_a$

3: $\Lambda = \begin{pmatrix} [& c, & g_a &] \\ [& 4N/g_a, & 0 &] \end{pmatrix}$

4: $k, u, v \leftarrow \mathrm{xgcd}(\mathrm{tr}(\alpha), \mathrm{tr}(\mathbf{i}\alpha))$

5: $q, w, - \leftarrow \mathrm{xgcd}(k, 4N)$

6: **if** $q \neq 1$: **return** $\perp$ ▷ α not generator.

7: **return** $\Lambda, wu \mod 4N, wv \mod 4N$ ▷ $\mathrm{tr}(\overline{(wu + \mathbf{i}wv)}\alpha_0) = 1 \mod 4N$

Algorithm 7 ComputeSmallSolutions

Require: $\Lambda(\alpha_0)$ a **reduced** lattice corresponding to solution $\sigma \in \mathbb{Z}[\mathbf{i}], \mathrm{tr}(\overline{\sigma}\alpha_0) = 0 \mod 4N$, λ such that $\gcd(\lambda, 2N) = 1$, (ρ_1, ρ_2) computed in ComputeLattice and finally d.

Ensure: (s, t) small solution of $2\lambda a_\alpha s + 2\lambda b_\alpha t = d \mod 4N$

1: $\mu \leftarrow \lambda^{-1} \mod 4N$

2: $v_{target} \leftarrow [-\mu \cdot d \cdot \rho_1 \mod 4N, -\mu \cdot d \cdot \rho_2 \mod 4N]$ ▷ $\mathrm{tr}(\overline{v_{target}}\lambda\alpha_0) = d \mod 4N$

3: $v_{close} \leftarrow \lfloor v_{target} \cdot \Lambda(\alpha_0)^{-1} \rceil \cdot \Lambda(\alpha_0)$

4: $(s, t) \leftarrow v_{target} - v_{close}$

5: **return** (s, t)

Algorithm 8 ReduceGivenMinima

Require: $\Lambda(\alpha)$ a generic lattice following Lemma 5 together with ω, one of its 2 minima.

Ensure: $\Lambda(\alpha)$ in its reduced form

1: $c, g_a, 4N/g_a \leftarrow \Lambda(\alpha)_{0,0}, \Lambda(\alpha)_{0,1}, \Lambda(\alpha)_{1,0}$

2: $a_1 \leftarrow \omega_1/g_a$

3: $a_2 \leftarrow (-c \cdot \omega_1 + g_a \cdot \omega_0)/(4N)$ ▷ $\omega = [a_1, a_2] \cdot \Lambda(\alpha)$

4: $1, a_4, -a_3 \leftarrow \mathrm{xgcd}(a_1, a_2)$ ▷ $a_1 a_4 - a_2 a_3 = 1$

5: $v_2 \leftarrow [a_3 \cdot c + a_4 \cdot (4N/g_\alpha), a_3 \cdot g_\alpha]$ ▷ $v_2 = [a_3, a_4] \cdot \Lambda(\alpha)$

6: $\mathbf{M} \leftarrow \begin{pmatrix} \omega \\ v_2 \end{pmatrix}$ ▷ $\mathbf{M}$ is a basis of $\Lambda(\alpha)$

7: $\Lambda(\alpha) \leftarrow$ OneStepReduction$(\mathbf{M})$

8: **return** $\Lambda(\alpha)$

Algorithm 9 OneStepReduction

Require: L is a 2×2 lattice basis where the first vector is already the first minimum.

1: $n_0 \leftarrow L_{0,0}^2 + L_{0,1}^2$

2: $m \leftarrow L_{0,0} \cdot L_{1,0} + L_{0,1} \cdot L_{1,1}$

3: $q \leftarrow \lfloor \frac{m}{n_0} \rceil$

4: $L_{\mathrm{ret}} \leftarrow \begin{pmatrix} L_{0,0} & L_{0,1} \\ L_{1,0} - q \cdot L_{0,0} & L_{1,1} - q \cdot L_{0,1} \end{pmatrix}$

5: **return** L_{ret}