

New algorithms for quaternion ideals in SQIsign

Antonin Leroux^{1,2} and Sina Schaeffler^{3,4}

¹ DGA-MI, Bruz, France

² IRMAR - UMR 6625, Université de Rennes, France

³ IBM Research Europe, Zurich, Switzerland

⁴ ETH Zurich, Zurich, Switzerland

Abstract. Many isogeny-based schemes rely on the Deuring correspondence and thus require computations with quaternion ideals. In this paper, we generalize the recent results of Leroux on the inert representation of quaternion ideals, yielding a normalized representation together with a set of simple and efficient algorithms for quaternion ideals in the context of the Deuring correspondence when the prime characteristic p is equal to 3 mod 4.

One of the main benefit of our new algorithms is that the size of the integers involved in the computations is both easy to bound, and smaller than previous work. In particular, when applied to the latest version of SQIsign, our algorithms yield a concrete integer bound of $O(p^4)$ in the worst case, proven under an experimentally verified assumption on the size of the integers required to perform the lattice reduction part of the response sampling. This bound improves upon all known previous bound by a factor at least $O(p^2)$.

Our new algorithms are used in the latest update of SQIsign's implementation submitted to the third round of the NIST call for additional post-quantum signatures. In particular, their low integer bounds facilitated the removal of a dependency on the GMP library in favor of a custom library of fixed-size integers without performance overhead.

We stress that our algorithms are not limited to SQIsign, but apply to any isogeny-based cryptographic protocol that relies on the Deuring correspondence such as the PRISM signature scheme for instance.

1 Introduction

Quaternion ideals play a central role in isogeny-based cryptography thanks to the Deuring correspondence. This correspondence enables the translation of problems in the realm of elliptic curves to problems in the realm of quaternion algebras where they can usually be solved more efficiently. Several recent post-quantum cryptographic schemes crucially rely on this paradigm. This is the case for SQIsign and all its variants [15, 7, 14, 27, 16, 10] and PRISM [4, 5]. These schemes are the most compact post-quantum signature schemes. While their performance

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

is not as good as some other families of schemes, it is sufficient to be considered practical. More recently, several more advanced protocols have been shown to be reachable from the Deuring correspondence such as Verifiable Random Functions, Ring signatures, Updatable encryption or signature with randomizable keys [24,11,26,6]. In many of cases, the performance profile of the solutions based on the Deuring correspondence is competitive with other post-quantum candidates.

Initially, the algorithms required to handle most of the operations required by the Deuring correspondence were considered as less critical than the elliptic curve and isogeny computations due to their efficiency and relative simplicity. However, as the field of isogeny-based cryptography has matured and schemes based on the Deuring correspondence become increasingly practical, these long-overlooked algorithms are becoming more critical.

In particular, the non constant-time implementation of SQIsign [2] submitted to second round of the NIST competition used generic algorithms to handle operations on quaternions ideals such as basis computation, ideal multiplication or ideal intersection. These algorithms were neither optimal for the size of the integers involved, nor for the number of operations required, forcing, in particular, the use of a big integer library such as GMP [31] in the implementation. This became problematic in subsequent implementations of SQIsign trying to progress toward constant-time either by replacing GMP, or by making the quaternion ideal operations constant-time [8,21,19,18,13].

In [25], Leroux shows that many quaternion ideals stemming from the Deuring correspondence have a very specific structure that can be exploited to obtain more efficient algorithms with small and provable bounds on the size of the integers involved. The potential gain from these algorithms is demonstrated on another application of the Deuring correspondence: the computation of modular polynomials [29,23]. While the application of these ideas to SQIsign is conjectured in [25], several required building blocks are missing there.

More precisely, Leroux looks at $\mathcal{O}_0$ -ideals of prime norm related to a special maximal order $\mathcal{O}_0$, which are the ideals most commonly met in the context of the Deuring correspondence. He shows that most of these ideals admit a basis called *in inert form*, that is amenable to efficient algorithms.

1.1 Contributions

The present work aims at bridging the gap left open in [25] by generalizing the inert ideal notion introduced for odd prime norm to all possible norms, and show that every $\mathcal{O}_0$ -ideal is the multiple of (and thus equivalent to) an inert ideal. This gives a normalized representation of all $\mathcal{O}_0$ -ideals that we show to be efficiently computable. This representation can also be exploited to derive generic and efficient intersection and multiplication algorithms. These operations are sufficient to handle most operations relevant to the Deuring correspondence, and we provide bounds on the size of the integers involved.

Then, we describe how to adapt SQIsign to make use of our new algorithms, and provide a bound on the integers required to instantiate the whole protocol. In

the worst case, this bound is in $O(p^4)$, under an experimentally verified assumption on the size of integers involved in the new lattice reduction step of SQIsign, and heuristically the bound can be a bit better than that with probability higher than $1 - 2^{-\lambda}$ where λ is the security level.

Our algorithms were used in the third round implementation of SQIsign [1] submitted to the NIST call for additional post-quantum signatures. This implementation uses fixed-size integers bounded by the bounds we provide in this work, and does not require any big integer library. The performance overhead of this modification is minor.

1.2 Related work

The first attempt to provide a fixed-size implementation of SQIsign and the Deuring correspondence was made by Kim, Lee, Kim and Lee in [19]. They prove an initial, very large bound on the integers involved in SQIsign without much optimizations to reduce this bound. Thus their bound is too large, not tight, and their implementation terribly slow. This was later improved by Kim, Lee and Yoo in [18] through the means of lattice reduction of many ideals appearing in SQIsign, and the substitution of HNF computations (incurring particularly large integers) by direct applications of LLL. However, to compute the multiplication and intersection of ideals required in SQIsign, this lattice reduction has to be applied on a family of 16 vectors (instead of 4 if a basis is computed otherwise as we propose). In the end, they obtain a worst case bound of $O(p^{6.5})$.

Very recently, Czuprynski, Mukherjee, and Sinha Roy [13] provided another attempt at provided bounds for SQIsign through a careful analysis of generic lattice algorithms. Their worst case bound appears to be $O(p^6)$ at best.

In comparison, our approach is to generalize Leroux’s idea of efficiently computing the HNF of inert ideals without too large integers by exploiting the specific structure of the quaternion ideals in the Deuring correspondence. By adding a reduction of some of the ideals appearing throughout the protocol to avoid a blow-up of their sizes, we obtain a smaller bound than other works [18,13]. Moreover, the integration of our new algorithms in the SQIsign codebase did not create a performance overhead, unlike the solution from [18] whose fixed-size implementation is approximately ten times slower (see [18, Table 4]). The direct comparison of both implementation might not be fair since the NIST implementation has probably been optimized more, but we believe that our approach outperforms other approaches thanks to its the smaller bounds and arguably simpler algorithms, and that it is currently the best way to implement the Deuring correspondence for prime characteristics equal to 3 mod 4.

Acknowledgements. We thank the SQIsign team for collaboration and feedback on the third round implementation of SQIsign. We thank Luca De Feo, and Eli Orvis for insightful discussions on this work. The work of Sina Schaeffler was supported by SNSF Consolidator Grant CryptonIs 213766.

Outline of the paper. In Section 2 we first give some background on quaternion ideals, and the integer sizes required by some building blocks our algorithms rely on. Section 3 then gives some structural results on quaternion ideals of a special order $\mathcal{O}_0$ and efficient for basic operations on these ideals. Section 4 builds on these to provide more advanced building blocks for applications to the Deuring correspondence. Finally, Section 5 details their application to SQIsign.

2 Background

In this section, we introduce the background on quaternion ideals and the Deuring correspondence necessary to follow this work, together with some useful notations and preliminary results.

For the rest of this work, a prime p equal to 3 mod 4 is considered fixed.

2.1 Quaternion ideals

We consider the quaternion algebra $B_{p,\infty}$ ramified at p and ∞ , which is a 4-dimensional vector space over the rational numbers $\mathbb{Q}$ with a basis given by $1, i, j, ij$ where multiplication is defined by $i^2 = -1, j^2 = -p, ij = -ji$. For $x = x_1 + x_i i + x_j j + x_{ij} ij \in B_{p,\infty}$ with $x_1, x_i, x_j, x_{ij} \in \mathbb{Q}$ the conjugate of x is $\bar{x} = x_1 - x_i i - x_j j - x_{ij} ij$, its norm $n(x) = x\bar{x}$ and its trace is $\text{tr}x = 2x_1$.

An order in $B_{p,\infty}$ is a rank 4 $\mathbb{Z}$ -lattice contained in $B_{p,\infty}$ which is also a subring. A maximal order is maximal for inclusion. What we call a quaternion ideal or simply ideal in this work is a left ideal of a maximal order. As a consequence, it is also a (rank 4) lattice. We only consider left ideals of maximal orders. These are also right orders of (possibly different) maximal orders. The norm $n(I)$ of an ideal I is the greatest common divisor (GCD) of the norms of its elements. We mostly consider cyclic or primitive ideals. An ideal I of an order $\mathcal{O}$ is cyclic if there is no $\ell \in \mathbb{Z}, \ell > 1$ such that $\frac{1}{\ell}I \subset \mathcal{O}$. Any quaternion ideal can be obtained from the order using two generators, one of which can be taken to be the norm. We usually call an ideal element generating the ideal together with the ideal norm a generator, even though most ideals are not principal. Finally, two ideals I, J are equivalent $I \sim J$ if there is $\alpha \in B_{p,\infty}$ non-zero such that $I = J\alpha$.

Definition 1. For any order $\mathcal{O} \subset B_{p,\infty}$, and we say an element $\gamma \in \mathcal{O}$ is primitive in $\mathcal{O}$ if the biggest positive integer ℓ such that $\gamma/\ell \in \mathcal{O}$ is 1. We say that γ is N -primitive in $\mathcal{O}$ if $\text{GCD}(\ell, N) = 1$.

When the order $\mathcal{O}$ is clear from the context (in this work, it is always a special order $\mathcal{O}_0$), we may only say that γ is primitive or N -primitive without explicit mention of $\mathcal{O}$. We define `MakePrimitive` to be the algorithm that takes a quaternion $\gamma \in \mathcal{O}_0$ and computes the primitive $\gamma' \in \mathcal{O}_0$ such that $\gamma = \mu\gamma'$ for some $\mu \in \mathbb{N}$.

We later use the following result on primitive generators of quaternion ideals.

Lemma 2. *Let $\mathcal{O}$ be quaternion order in $B_{p,\infty}$. For any integer $N > 0$ coprime to p , if $\gamma \in \mathcal{O}$ is N -primitive and satisfies $n(\gamma) = 0 \pmod{N}$, then $I = \mathcal{O}\langle\gamma, N\rangle$ is a left cyclic $\mathcal{O}$ -ideal of norm N .*

Proof. We recall that for any ideal I of $\mathcal{O}$, we have that $n(I) = \text{GCD}(\{n(\alpha), \alpha \in I\})$. It is clear that I is an ideal and that its norm divides N^2 . To prove that its norm is N exactly we must prove that there exists an element $\alpha \in I$ such that $\text{GCD}(n(\alpha), N^2) = N$. If γ satisfies this property then the proof is done, otherwise let us write $N' \mid N$ such that $\text{GCD}(n(\gamma), N^2) = NN'$. Then, there exist $N_1, N_2 \in \mathbb{N}$ such that $\text{GCD}(N_2, N') = 1$, $N' \mid N_1$ and $N = N_1 N_2$.

For any prime $M \neq p$ and primitive $\alpha \in \mathcal{O}$, it is easily verified (for instance since there existing a trace preserving (modulo M) isomorphism between $\mathcal{O}/M\mathcal{O}$ and $\mathcal{M}_2(\mathbb{Z}/M\mathbb{Z})$) that

$$\alpha \in M\mathcal{O} \Leftrightarrow \text{tr}(\alpha\beta) = 0 \pmod{M}, \quad \forall \beta \in \mathcal{O}.$$

Thus, since γ is N -primitive, there exists β such that $\text{tr}(\gamma\beta)$ is invertible modulo N . Then, there exists $\lambda \in \mathbb{Z}/N\mathbb{Z}^\times$ such that $\gamma + \lambda N\beta$ is the element we're looking for. Indeed, $n(\gamma + \lambda N\beta) = n(\gamma) + N\text{tr}(\gamma\beta) + N^2 n(\beta)$. It then suffices that $n(\gamma)/N + \lambda \text{tr}(\gamma\beta)$ is invertible mod N . As $n(\gamma)/N$ is not invertible mod N while $\text{tr}(\gamma\beta)$ is, such a λ can always be found for any N . $\square$

As explained in Section 2.2, the Deuring correspondance most often requires to work with ideal classes rather than ideals. Therefore, in most cases any representative of a given class can be used. We therefore often search for and then use a representative more convenient for computation for instance because the norm is smaller, or coprime to another integer. It is a well established result since [20] that two ideals $I \sim J$ satisfy $J = I \frac{\beta}{n(I)}$ where $\beta \in I$ has norm $n(I) \cdot n(J)$. The following result shows that $\overline{\beta}$ is also a generator of J .

Lemma 3. *Let I be a left primitive $\mathcal{O}$ -ideal, and let β be a primitive element of I . Then $J = I \frac{\beta}{n(I)} = \mathcal{O}\langle\overline{\beta}, \frac{n(\beta)}{n(I)}\rangle$.*

Proof. If β is primitive, then by Lemma 2 we get that $I = \mathcal{O}\beta + \mathcal{O}n(I)$, this means that $J = \mathcal{O} \frac{\beta\overline{\beta}}{n(I)} + \mathcal{O}\overline{\beta} = \mathcal{O}\langle\overline{\beta}, \frac{n(\beta)}{n(I)}\rangle$ $\square$

2.2 The Deuring correspondence

As this work focuses almost exclusively on quaternions, we skip the introduction of elliptic curves, and isogenies that are only briefly mentioned in the present section, and we refer the reader to [30] for background and definitions on these objects. For a complete treatment of the Deuring correspondence focused on the application to isogeny-based cryptography, we also refer the reader to [22].

The Deuring correspondence is often seen as a dictionary between two worlds, each element of one world corresponding to an element from the other world (up to isomorphisms and various notions of equivalence). Concretely, each

supersingular curve E over $\mathbb{F}_{p^2}$ has its endomorphism ring $\text{End}(E)$ isomorphic to some maximal order $\mathcal{O}$ in $B_{p,\infty}$. This gives a way to associate isomorphism classes of maximal orders in $B_{p,\infty}$ to isomorphism classes of supersingular curves over $\mathbb{F}_{p^2}$ (up to Galois conjugacy). Then, $\text{Hom}(E_1, E_2)$ is isomorphic to left $\mathcal{O}_1$ and right $\mathcal{O}_2$ ideals, and each isogeny $\varphi : E_1 \rightarrow E_2$ can be associated to some left $\mathcal{O}_1$ -ideal I_φ isomorphic to $\text{Hom}(E_2, E_1) \circ \varphi$ through the notion of kernel ideal:

$$I_\varphi := \{\alpha \in \text{End}(E_1) : \alpha(\ker \varphi) = 0\}$$

The norm of I_φ is equal to the degree of φ .

Another way of considering the Deuring correspondence, closer to the original formulation of the equivalence of categories of the Deuring correspondence (see [32, Theorem 42.3.2]), is to fix some initial maximal order $\mathcal{O}_0$ isomorphic to the endomorphism ring of some supersingular curve E_0 , and consider the bijection⁵ between isomorphism classes of supersingular curves, and equivalence classes of $\mathcal{O}_0$ -ideals. From that point of view, it is clear that most of the relevant computations can be performed on $\mathcal{O}_0$ -ideals. We follow this approach in this work by optimizing all operations involving $\mathcal{O}_0$ -ideals for some natural choice of $\mathcal{O}_0$ introduced below.

The special maximal order $\mathcal{O}_0$. The quaternion algebra $B_{p,\infty}$ has a maximal order $\mathcal{O}_0$ of $\mathbb{Z}$ -basis $\langle 1, i, \frac{i+j}{2}, \frac{1+k}{2} \rangle$ with $i^2 = -1$, $j^2 = -p$ and $k = ij$. $\mathcal{O}_0$ is particularly suitable for some computations, because it contains i that is of norm 1, and orthogonal to j . This order is isomorphic to the endomorphism ring of the supersingular elliptic curve $E_0/\mathbb{F}_{p^2}$ of j -invariant 1728, given by $y^2 = x^3 + x$. Moreover, there are explicit endomorphisms π and ι such that $\text{End}(E_0) = \langle 1, \iota, \frac{\iota+\pi}{2}, \frac{1+\iota\pi}{2} \rangle$, where π is the Frobenius isogeny and ι is the map $(x, y) \mapsto (-x, \sqrt{-1}y)$. We fix the notation of $\mathcal{O}_0$ for the rest of this article.

Algorithmic applications, and useful operations on quaternion ideals. In the concrete applications of the Deuring correspondence, the most interesting feature is the ability to perform isogeny operations on the quaternion side of the Deuring correspondence. In this context, two operations on quaternion ideals naturally appears as critical: multiplication and intersection. The former is the quaternion analog of the composition of isogenies, and allows us to cover every isomorphism class of isogenies by considering $\varphi_1 : E_0 \rightarrow E_2$ and $\varphi_2 : E_0 \rightarrow E_2$, and $\psi = \hat{\varphi}_2 \circ \varphi_1 : E_1 \rightarrow E_2$ whose corresponding ideal is equal to $I_\psi = \overline{I}_1 \cdot I_2$.

In this case, if the degrees of φ_1 and φ_2 are coprime, one can define the commutative pushforward diagram (sometimes called isogeny diamond, or SIDH diagram), corresponding to the two possible decomposition of the isogeny $\psi : E_0 \rightarrow E_3$ of degree $\deg \varphi_1 \cdot \deg \varphi_2$ equal either to $[\varphi_1]_* \varphi_2 \circ \varphi_1$ or to $[\varphi_2]_* \varphi_1 \circ \varphi_2$. This diagram is used almost everywhere in isogeny-based cryptography. The ideal corresponding to ψ is equal to $I_\psi = I_1 \cap I_2$.

The final useful operation is the computation of a representation (typically a basis) of I from a set of generators α, N such that $I = \mathcal{O}_0 \langle \alpha, N \rangle$. This is

⁵ that can be made functorial by considering isogenies and morphisms of Hom-modules

useful to compute the kernel ideal of φ (as the generator α corresponds to an endomorphism such that $\ker \varphi = \ker \alpha \cap E_0[N]$), or to compute equivalence of ideals with Lemma 3.

2.3 Size of integers used in algorithms, and known building blocks.

One of the main goals of this work is to obtain a precise bound on the size of the integers involved in each of the algorithm introduced here. Of course, this covers the size of the input and outputs, but also the size of the temporary variables that may occur during the computation of the algorithm. Hereafter, we call a bound on the maximum of any integer involved in the computation of some algorithm A , a **working bound** of A . All our results, thus include bounds on the output, and the working size of each of our algorithms.

In the working bounds that we encounter throughout this work, the function $\frac{p+8x^2}{2x}$ appears often. This motivates the definition of

$$\text{WB}(x) \mapsto \frac{p+8x^2}{2x}$$

where the parameter p is the one fixed at the beginning of Section 2.

In Section A.1, we introduce some bounds for various classical algorithm from the litterature that we will use as basic building blocks for our algorithms.

3 Fast algorithms for $\mathcal{O}_0$ -ideals

This section introduces our main algorithms for efficient computation on quaternion ideals for the Deuring correspondence. Section 3.1 introduces the main structural results upon which the rest of the algorithms of this section rely.

3.1 Structural results on $\mathcal{O}_0$ -ideals

Our goal in this section, is to introduce a canonical way of representing $\mathcal{O}_0$ -ideals. The main ingredient for that are the inert ideals.

Inert ideals. Our notion of inert ideals is theavily inspired by the structural result [25, Lemma 3.1] on $\mathcal{O}_0$ -ideals of odd prime norm which we partially restate below as Definition 4 (with some small change of notations).

Definition 4. *Let $p \equiv 3 \pmod{4}$ and N be odd. Then, we say a left $\mathcal{O}_0$ -ideal of norm N admits a basis⁶ in inert form if it has a basis of the form:*

⁶ the rows of the matrix below are the coefficients of the basis elements of the ideal in the basis $\langle 1, i, j, k \rangle$ of $B_{p,\infty}$

$$M_{\text{inert}}^N(x, y) = \begin{pmatrix} N & 0 & 0 & 0 \\ 0 & N & 0 & 0 \\ x & \frac{1}{2} + y & \frac{1}{2} & 0 \\ -y - \frac{1}{2} & x & 0 & \frac{1}{2} \end{pmatrix}, \quad (1)$$

where $x, y \in \mathbb{N}$ are strictly smaller than N .

When N is an odd prime, Leroux showed that there are at most two ideals of norm N that cannot have a basis in inert form. Moreover, these two special ideals must be principal.

When N is composite, not all non-principal ideals have a basis of this form. However, we are going to see later in Lemma 7 that every ideal of norm N is equivalent to an ideal of that form (for an N' dividing N). This motivates to introduce a formal definition for these ideals.

Definition 5. Let $N \in \mathbb{N}^*$ and I be a left cyclic $\mathcal{O}_0$ -ideal of norm N . We say that I is **in inert form**, or simply **inert** if it contains an element of the form $x + iy + (i + j)/2$ for some $0 \leq x, y < N$.

As exhibited first in [25], inert ideals are very convenient for practical computation. For instance, the intersection of inert ideals is also inert and very efficient to compute. We see in Section 3.3 that multiplication of two inert ideals is also very efficient and yields an ideal that is almost in inert form (but not an $\mathcal{O}_0$ -ideal). The definition of inert ideals naturally yields a very compact representation.

Lemma 6. For I and $\mathcal{O}_0$ -ideal of norm N . The following are equivalent:

1. I is inert.
2. I admits a basis of the form $M_{\text{inert}}^N(x, y)$.

Moreover, there exists a unique pair of integers $0 \leq x, y < N$ such that $x + iy + (i + j)/2$ is in I .

Proof. First, since I is an $\mathcal{O}_0$ -ideal of norm N it must contain N . Then, since we can always multiply by i on the left, the existence of an element of the form $(j + i)/2 + yi + x$ imply that a basis of a sublattice of I is given by $M_{\text{inert}}^N(x, y)$. Equality follows from the equality of discriminants.

Regarding unicity of the choice of $0 \leq x, y < N$, assume there exists two distinct elements $x + iy + (i + j)/2$ and $x' + iy' + (i + j)/2$ in I . By subtraction we get that $(x - x') + i(y - y')$ is also contained in I . By construction, this is an element of $\mathbb{Z}[i]$ which is non-zero mod $N\mathbb{Z}[i]$. Given that $M_{\text{inert}}^N(x, y)$ is a basis of I , it is not possible to have an element in $\mathbb{Z}[i] \cap I \neq 0 \pmod{N\mathbb{Z}[i]}$. This concludes the proof. $\square$

Lemma 6 proves that the triple $N, (x, y)$ with $0 \leq x, y < N$ is a valid representation for inert ideals of norm N . Thus, in the rest of this paper, we often implicitly consider inert ideals as tuple of integers, and sometimes denote the ideal of basis $M_{\text{inert}}^N(x, y)$ as simply $N, (x, y)$.

Generic ideals. As already explained, not all $\mathcal{O}_0$ -ideals are in inert form (for instance the split ideals from [25, Lemma 3.1]). This section therefore deals with the generic case.

Generally speaking, we are always able to factor an ideal I as $I_1\alpha_0$ where $\alpha_0 \in \mathbb{Z}[i]$ and I_1 is in inert form. The element α_0 is sometimes called the **split part** of I . This is proven in Lemma 7 below. Note that the generic case is also treated in [12,3] but in a different manner that is less amenable to concrete computations. In particular, the ideals I and I_1 are equivalent which means that in case we are interested in the class of I rather than a specific representative of the class, we can use I_1 instead of I .

Lemma 7. *Let $p = 3 \pmod{4}$, let I be an $\mathcal{O}_0$ -ideal of norm N , there exists a unique (up to multiplication by $\text{Aut}(\mathcal{O}_0)$ on the left) element $u + iv \in \mathbb{Z}[i]$, and ideal I_1 is inert form such that*

$$I = I_1 \cdot (u + iv)$$

Proof. Let $u + iv = \alpha_0$ be the biggest⁷ element in $\mathbb{Z}[i]$ such that $I \subset \mathcal{O}_0\alpha_0$. Up to multiplication by $\text{Aut}(\mathcal{O}_0) = \mathbb{Z}[i]^\times$, the element α_0 is well-defined.

Since $I \subset \mathcal{O}_0\alpha_0$, we get that $I_1 = I\alpha_0^{-1} = I\overline{\alpha_0}/n(\alpha_0)$ is an $\mathcal{O}_0$ integral ideal. Let us take some generator γ of I_1 . It remains to prove that I_1 is in inert form. In the proof of Lemma 8, we will see that the cases where Algorithm 1 fail (and equivalently the cases where I is not inert by Lemma 8) are the cases where the generator $\gamma = (u + iv) + j(c + id)$ is such that $(u + iv)$ and $(c + id)$ are both contained in a common $\mathbb{Z}[i]$ -ideal (which is principal). Thus, if we assume that I_1 is not inert, then there exists an element $\alpha'_0 \in \mathbb{Z}[i]$ of norm N' bigger than 1 (and dividing N) such that $\gamma = \gamma'\alpha'_0$. Since $I_1 = \mathcal{O}_0\gamma + \mathcal{O}_0n(I_1)$, and $N' = \overline{\alpha'_0}\alpha'_0$, it is clear that $I_1 = I'_1\alpha'_0$ which contradicts the maximality of α_0 . Thus I_1 must be inert. $\square$

3.2 Generator to ideal, the inert case

First, we need to compute the inert representation of an ideal. As justified in Section 2.2, most often when the inert representation is not already known, the ideal is given as a tuple γ, N such that $I = \mathcal{O}_0\langle\gamma, N\rangle$. To ensure that all ideals we manipulate are always in inert representation, we need to go from the generators γ, N to the inert representation. This is the goal of this section. The only assumptions we make on γ is that it is N -primitive and that its norm is divisible by N . If it turns out that $\mathcal{O}_0\langle\gamma, N\rangle$ is not an inert ideal, the algorithm outputs $\perp$. Since we are in $\mathcal{O}_0 \subset \mathbb{Z}/2\langle i, j \rangle$, the case of even norm is a bit more complicated to handle. For this reason, we start with the odd case.

The odd case. Our algorithm to go from generator to ideal in the odd case is called **InertGenToIdealOdd** and it is detailed as Algorithm 1 below. It is almost

⁷ for the usual norm $n(u + iv) = u^2 + v^2$

identical to [25, Algorithm 2] which works for prime N . The generalization requires the following tweak: when the first invertibility check of $c^2 + d^2$ modulo N fails, instead of multiplying the provided generator γ by $(t + j)$ for some t such that $t^2 + p$ is non-zero mod N , we multiply it by $N_2 + j$ for some well-chosen $N_2 | N$. We prove correctness of Algorithm 1 with Lemma 8 which reuses part of the proof of [25, Lemma 3.1]. We explain there why the new tweak works. Without loss of generality, we assume that the generator γ is in $\mathbb{Z}\langle 1, i, j, k \rangle$. Indeed γ is in $\mathcal{O}_0 \subset \mathbb{Z}/2\langle 1, i, j, k \rangle$ and 2γ is also a generator of the same ideal as the norm is odd.

Algorithm 1 InertGenToldealOdd

Input: an odd number N , and an N -primitive element $\gamma = (a + ib + (c + id)j)$ where $a, b, c, d \in \mathbb{Z}$ and $n(\gamma) = 0 \pmod N$

Output: $\perp$ or (x, y) , the inert ideal of norm N generated by γ

- 1: $s \leftarrow c^2 + d^2 \pmod N$
- 2: **if** $\text{GCD}(s, N) \neq 1$ **then**
- 3: Let N_2 be the greatest divisor of N coprime to s .
- 4: $\gamma \leftarrow (N_2 + j)\gamma$, and update a, b, c, d accordingly.
- 5: $a, b, c, d \leftarrow a, b, c, d \pmod N$
- 6: $s \leftarrow c^2 + d^2$
- 7: **if** $\text{GCD}(s, N) \neq 1$ **then**
- 8: return $\perp$
- 9: $t \leftarrow (2s)^{-1} \pmod N$
- 10: $x \leftarrow (ac + bd)t \pmod N$
- 11: $y \leftarrow (bc - ad - s)t \pmod N$
- 12: return (x, y)

Lemma 8. *Algorithm 1 outputs $\perp$ if and only the cyclic $\mathcal{O}_0$ -ideal I do not admit a basis in inert form.*

Moreover, the output of Algorithm 1 are upper-bounded by N , and a working bound for InertGenToldealOdd is N^2 .

Proof. The goal of the algorithm is to find the element of β of the form $x + iy + \frac{i+j}{2}$ in the ideal (if it exists). Following the method described in [25, Lemma 3.1], the idea is to find this β from another element $\gamma = a + ib + (c + id)j = (a + ib) + j(c - id)$ where $c^2 + d^2$ is invertible mod N as one then gets a suitable β as $(c - id)t\gamma \pmod N\mathcal{O}_0$ where $t = (2(c^2 + d^2))^{-1} \pmod N$. This is exactly what is done in the algorithm as soon as the suitable γ has been found. This proves that if the algorithm succeeds, then the ideal $\mathcal{O}_0\langle \gamma, N \rangle$ must be inert.

Now, we just need to justify that when I is inert, the algorithm succeeds in finding this γ .

We start naturally with the generator γ in input. If $s = c^2 + d^2$ is not coprime to N , then taking N_2 to be the greatest divisor of N coprime to s and using $(N_2 + j)\gamma$ works. We have $(N_2 + j)\gamma = a' + ib' + j(c' - id')$ where

$c' - id' = ((a + ib) + N_2(c - id))$. First it is clear that $c'^2 + d'^2$ is coprime to N_2 as it is equal to $a^2 + b^2 \pmod{N_2}$ and $a^2 + b^2$ must be coprime to N_2 as otherwise $c^2 + d^2$ would not be coprime to N_2 as $\mathfrak{n}(\gamma) = 0 \pmod{N_2}$. Now, let us take any prime q dividing $\text{GCD}(s, N)$, we must have $a^2 + b^2 = 0 \pmod{q}$ as q divides both $c^2 + d^2$ and $\mathfrak{n}(\gamma) = a^2 + b^2 + p(c^2 + d^2)$. So $a + ib$ and $c - id$ are both contained in a $\mathbb{Z}[i]$ -ideal of norm q . If both contained in the same ideal of norm q , they are both divisible by some generator $\alpha_0 \in \mathbb{Z}[i]$ of this ideal (as all the ideals are principal in $\mathbb{Z}[i]$) and so γ is divisible (on the right) by α_0 (and N too as $q = \overline{\alpha_0}\alpha_0$ divides N). So this means that $I = I'\alpha_0$ where I' has norm $N' = N/q$ which implies that $N'\alpha_0$ is contained in I which is not possible from the shape of the basis if I is inert. Thus $a + ib$ and $c - id$ cannot be contained in the same ideal.

By absurd, we deduce from this that $c' - id' = a + ib + N_2(c - id)$ cannot have its norm divisible by q . Assume that q divides $\mathfrak{n}(c' - id')$. Then $c' - id'$ is contained in one of the two $\mathbb{Z}[i]$ -ideals of norm q which implies that either $a + ib$ or $(c - id)$ is contained in both $\mathbb{Z}[i]$ -ideals of norm q (the factor N_2 can be removed as it is coprime to q). Then $(a + ib)$ and $(c - id)$ are both contained in the same ideal of norm q (as there are only two such ideals) which is impossible as established above.

Thus $c'^2 + d'^2$ is also coprime to any prime divisor of s . As it is also coprime to N_2 , it is coprime to all prime divisors of N and thus to N . Therefore if I is inert, the algorithm succeeds.

Since one can reduce the generator $\gamma \pmod{N\mathcal{O}_0}$ without changing the result of the algorithm, we can assume without loss of generality that γ is reduced, and thus that a, b, c, d are smaller than N . Then, all operations modulo N require to manipulate numbers smaller than N^2 before reduction. The computation of N_2 can be done using iterated GCD computations on number smaller than N which can be done with numbers smaller than N by Lemma 25. The values x, y can be reduced modulo N and thus are upper-bounded by N . $\square$

Power of two inert case. This special case needs to be handled slightly differently. First, we give a condition to detect inert ideals of even norm with Lemma 9. The proof of this result underlies Algorithm 2.

Lemma 9. *Let e, γ such that $\mathcal{O}_0\langle 2^e, \gamma \rangle$ is a cyclic $\mathcal{O}_0$ -ideal of norm 2^e , then Algorithm 2 succeeds if and only if I is inert.*

Proof. Let us take $\gamma = t + ix + \frac{i+j}{2}y + \frac{1+k}{2}z$ a generator of I . Without loss of generality, we can assume that γ is primitive in $\mathcal{O}_0$ since I is cyclic.

Let us now show that if I is inert, then the algorithm succeeds. The algorithm fails when either $(y, z) = (0, 0) \pmod{2}$ or $(y, z) = (1, 1) \pmod{2}$ and $x = t \pmod{2}$. We now show that these two cases are not possible when I is inert. We rely on the fact that if I is inert, then $I + 2\mathcal{O}_0$ cannot contain $1 + i$ (which follows from the shape of the basis of inert ideals).

For the first case, let us assume by absurd that $y, z = 0, 0 \pmod{2}$. Then, $\gamma = t + xi \pmod{2\mathcal{O}_0}$, $t, x \neq 0, 0 \pmod{2}$ since γ is primitive. If $t, x = 1, 0 \pmod{2}$

or $0, 1 \bmod 2$, then $1 \in I + 2\mathcal{O}_0$ or $i \in I + 2\mathcal{O}_0$ which is not possible since $I + 2\mathcal{O}_0$ is an ideal of norm 2. Finally, if $t, x = 1, 1 \bmod 2$ then $i + 1 \in I + 2\mathcal{O}_0$ which is impossible as I is inert.

Again par absurd, assume that $(y, z) = (1, 1) \bmod 2$ and $x = t \bmod 2$. Then,

$$\gamma(1 + i) = (t - x) + i(t + x) + \frac{i + j}{2}(y + z) + \frac{1 + k}{2}(z - y).$$

Since $y = z = 1 \bmod 2$, $\gamma(1 + i) = (t - x) + i(t + x) \bmod 2\mathcal{O}_0$. And so if $t = x \bmod 2$, then $\gamma(1 + i) \in 2\mathcal{O}_0$ which implies that $\gamma \in \mathcal{O}_0(1 - i) = \mathcal{O}_0(1 + i)$ (since $2 = (1 - i)(1 + i)$), which implies that $I + 2\mathcal{O}_0$ contains $1 + i$ which is again impossible if I is inert.

This concludes the proof that the algorithm succeeds when I is inert. Next we prove that when the algorithm succeeds, I is inert, and the result is correct.

Let us first show that if the algorithm succeeds, then after the if conditions, γ is a generator of I and $y, z \bmod 2 \in \{(0, 1), (1, 0)\}$. Either this is true from the start, or we initially have $(y, z) = (1, 1) \bmod 2$ and $x \neq t \bmod 2$. Then,

$$\frac{i + j}{2}\gamma = \frac{p + 1}{4}(y + zi) + \frac{i + j}{2}t - \frac{1 + k}{2}x$$

Since one of x or t is non zero modulo 2, after replacing γ by $(i + j)\gamma/2$ the updated y, z satisfy the desired condition. Moreover $\gamma' = (i + j)\gamma/2$ is 2-primitive (as one of x or t is non zero modulo 2) in $\mathcal{O}_0$ and so $I' = \mathcal{O}_0\langle\gamma', 2^e\rangle$ is an ideal of norm 2^e by Lemma 2. Moreover, $I' \subset I$ since $\frac{i + j}{2}$ is in $\mathcal{O}_0$ and thus $I = I'$ by equality of norms. So the new γ is also a generator of I .

Let us now assume that $y, z \bmod 2 \in \{(0, 1), (1, 0)\}$. We get that $y^2 + z^2 = 1 \bmod 2$ and therefore is invertible modulo 2^e . Thus:

$$(y - iz)\gamma = (y(t + \frac{z}{2}) + z(x + \frac{y}{2})) + i(y(x + \frac{y}{2}) + z(t + \frac{z}{2})) + j\frac{z^2 + y^2}{2}$$

And so by multiplying by $(z^2 + y^2)^{-1} \bmod 2^e$ and reducing modulo $2^e\mathcal{O}_0$ we get a basis element of the correct form which proves that I is inert. As these are exactly the operation performed by the algorithm, this also proves that the algorithm is correct. $\square$

Lemma 10. *The output of Algorithm 1 are upper-bounded by 2^e , and a working bound for `InertGenToldealPowTwo` is 2^{2e} .*

Proof. Without loss of generality, we may assume that the values t, x, y, z are contained in $[0, 2^e[$. The proof is identical to the second part of the proof of Lemma 8. $\square$

3.3 Intersection and multiplication and intersection of coprime inert ideals

The intersection of inert ideals of coprime norm is particularly simple as explained in [25]. Indeed if I_1 is given by $(x_1, y_1), N_1$ and I_2 by $(x_2, y_2), N_2$ then

$$I_1 \cap I_2 = (\text{CRT}([x_1, x_2], [N_1, N_2]), \text{CRT}([x_1, x_2], [N_1, N_2])), N_1 N_2.$$

Algorithm 2 InertGenToldealPowTwo Ideal basis computation of norm 2^e in inert form

Input: Exponent e , and a 2-primitive element $\gamma = t + xi + y\frac{i+j}{2} + z\frac{1+k}{2}$ such that $n(\gamma) = 0 \pmod{2^e}$

Output: $\perp$ or (X, Y) the inert ideal of norm 2^e generated by γ

```

1: if  $y, z \pmod{2} \notin \{(0, 1), (1, 0)\}$  then
2:   if  $y, z \pmod{2} = (1, 1)$  and  $x \not\equiv t \pmod{2}$  then
3:      $\gamma \leftarrow \frac{i+j}{2}\gamma \pmod{2^e\mathcal{O}_0}$  and update  $t, x, y, z$  so that  $\gamma = t + xi + y\frac{i+j}{2} + z\frac{1+k}{2}$ .
4:   else
5:     Return  $\perp$ 
6:    $s \leftarrow y^2 + z^2 \pmod{2^e}$ 
7:    $s \leftarrow s^{-1} \pmod{2^e}$ 
8:    $X \leftarrow t(yz + zy + yz) \pmod{2^e}$ 
9:    $Y \leftarrow s(yx - zt - z^2) \pmod{2^e}$ 
10: return  $(X, Y)$ 

```

Multiplication of inert ideals was not covered in [25]. Our new algorithm is introduced later as Algorithm 3. It follows directly from the following result and its proof.

Lemma 11. *Let $I_1 = (x_1, y_1)$, $I_2 = (x_2, y_2)$ be two inert left $\mathcal{O}_0$ -ideals of coprime norm N_1, N_2 . Then, the multiplication $K = \overline{I_1} \cdot I_2$ admits the following basis*

$$M_{\times, \text{inert}}^{N_1 N_2}(X_1, X_2, Y_1, Y_2) = \begin{pmatrix} N_1 N_2 & 0 & 0 & 0 \\ 0 & N_1 N_2 & 0 & 0 \\ X_1 + X_2 & \frac{Y_2 - Y_1}{2} & \frac{1}{2} & 0 \\ -\frac{Y_1 + Y_2}{2} & X_2 - X_1 & 0 & \frac{1}{2} \end{pmatrix}$$

for some $X_1, X_2, Y_1, Y_2 \in \mathbb{N}$ with $0 \leq X_1, X_2 \leq N_1 N_2$ and $0 \leq Y_1, Y_2 \leq 2N_1 N_2$.

Proof. We give a constructive proof that also underlies our algorithm to compute the basis of K .

The lattice generated by the basis $M_{\times, \text{inert}}^{N_1 N_2}(X_1, X_2, Y_1, Y_2)$ is clearly of rank 4 and has the correct discriminant for K . Thus, it suffices to prove that it is contained in K .

First, as $N_1 N_2$ is the norm of K , $N_1 N_2 \mathcal{O}_0$ is contained inside K . Thus, it is clear that $N_1 N_2$ and $N_1 N_2 i$ are contained in K .

Since N_1, N_2 are coprime, there are two integers u, v such that $uN_1 - vN_2 = 1$.

Let us write $\alpha_\ell = x_\ell + iy_\ell + (i+j)/2$ for $\ell \in \{1, 2\}$. Then $vN_2 \overline{\alpha_1} + uN_1 \alpha_2$ is contained in K and can be reduced mod $N_1 N_2 \mathcal{O}_0$ as $(X_1 + X_2) + i(Y_2 - Y_1)/2 + j/2$ where $X_1 = vN_2 x_1 \pmod{N_1 N_2}$, $X_2 = uN_1 x_2 \pmod{N_1 N_2}$ and $Y_1 = vN_2(2y_1 + 1) \pmod{2N_1}$, and $Y_2 = uN_1(2y_2 + 1) \pmod{2N_2}$. Clearly we can choose $0 \leq X_1, X_2 \leq N_1 N_2$ and $0 \leq Y_1, Y_2 \leq 2N_1 N_2$.

The final basis element is obtained similarly from $vN_2 i \overline{\alpha_1} + uN_1 i \alpha_2$. One can easily verify that this element is equal to $-(Y_1 + Y_2)/2 + i(X_2 - X_1) + k/2$ $\square$

Algorithm 3 InertIdealMultiplication

Input: coprime integers N_1, N_2 , and two inert ideals $I_1 = (x_1, y_1)$ and $I_2 = (x_2, y_2)$
Output: $\perp$ or a representation for $K = \overline{I_1} \cdot I_2$

- 1: $d, u_1, u_2 \leftarrow \text{XGCD}(N_1, N_2)$ // $u_1 N_1 + u_2 N_2 = d$
- 2: **if** $d \neq 1$ **then**
- 3: return $\perp$
- 4: $X_1 \leftarrow N_2(-u_2 x_1 \bmod (N_1))$
- 5: $X_2 \leftarrow N_1(u_1 x_2 \bmod (N_2))$
- 6: $Y_1 \leftarrow N_2(-u_2(2y_1 + 1) \bmod (2N_1))$
- 7: $Y_2 \leftarrow N_1(u_1(2y_2 + 1) \bmod (2N_2))$
- 8: return (X_1, X_2, Y_1, Y_2)

Lemma 12. *For two ideals in input of respective norm N_1, N_2 , the integers in the output of Algorithm 3 are upper-bounded by $2N_1 N_2$. The working bound of **InertIdealMultiplication** is $4 \max(N_1^2, N_2^2)$.*

Proof. By Lemma 25, we get that u_1, u_2 are smaller than $\max(N_1, N_2)$ and that the working bound for XGCD is the same. The final values X_1, X_2, Y_1, Y_2 are clearly smaller than $2N_1 N_2$. The multiplications mod $2N_i$ require to handle values that can be upper-bounded by $4N_i^2$. $\square$

3.4 Generator to ideal, the generic case

Let us now describe a generator to ideal algorithm for generic $\mathcal{O}_0$ -ideals. Following Lemma 7, the idea is first to identify the split part, factor it out, and reduce to the inert case.

Algorithm 4 GenToIdeal

Input: an integer N , an N -primitive element $\gamma = (a + ib + j(c - id))$ where $a, b, c, d \in \mathbb{Z}$ and $n(\gamma) = 0 \bmod N$
Output: $u + iv, (x, y)$ such that $I = I_1 \cdot (u + iv)$ where $I_1 = (x, y)$

- 1: $u + iv \leftarrow \text{GaussGCD}(a + ib, c - id)$
- 2: $u + iv \leftarrow \text{GaussGCD}(u + iv, N)$
- 3: **if** $u + iv = \pm i$ **then**
- 4: $u + iv \leftarrow 1$
- 5: $N_2 \leftarrow u^2 + v^2$
- 6: $N_1 \leftarrow N/N_2$
- 7: $\gamma \leftarrow \gamma^{\frac{u-iv}{N_2}} \bmod N_1 \mathcal{O}_0$.
- 8: Let e be the biggest exponent such that 2^e divides N_1 .
- 9: $x_1, y_1 \leftarrow \text{InertGenToIdealOdd}(N_1/2^e, \gamma)$
- 10: $x_2, y_2 \leftarrow \text{InertGenToIdealPowTwo}(2^e, \gamma)$
- 11: $x \leftarrow \text{CRT}([x_1, x_2], [N_1/2^e, 2^e])$
- 12: $y \leftarrow \text{CRT}([y_1, y_2], [N_1/2^e, 2^e])$
- 13: return $(N_1, (x, y)), u + iv$

Lemma 13. *Algorithm 4 is correct.*

Moreover, assuming that the values of a, b, c, d are reduced mod N , the bound on the integers given in output is N , and the working bound of **GenToldeal** is N^2 .

Proof. By the proof of Lemma 7, the factor $u + iv$ we're looking for is the biggest $u + iv$ such that $I \subset \mathcal{O}_0(u + iv)$. We're going to prove that this $u + iv$ must be equal to $\alpha = \text{GCD}_{\mathbb{Z}[i]}(a + ib, c - id, N)$.

First since $I \subset \mathcal{O}_0(u + iv)$, the input γ must be equal to $\gamma'(u + iv)$, and N is also divisible by $u + iv$. This proves that $(u + iv)$ divides α . Conversely, $\alpha = \text{GCD}_{\mathbb{Z}[i]}(a + ib, c - id, N)$ divides γ and N , and so $\mathcal{O}_0\gamma + \mathcal{O}_0N \subset \mathcal{O}_0\alpha$. By definition of $u + iv$ we have $\mathcal{O}_0\alpha \subset \mathcal{O}_0(u + iv)$ (as otherwise α would be bigger than $u + iv$) and this proves the correctness of the computation of $u + iv$.

Then since the inert part is obtained by multiplying by $(u + iv)^{-1}$, we get that the inert part is equal to $I_1 := \mathcal{O}_0(\gamma(u - iv)/N_2, N_1)$. The computation of I_1 by calling it on its inert power of two and odd parts before intersecting with the CRT is then correct.

Since $u^2 + v^2 = N_1$ which divides N , we must have that at then end $|u, v| \leq \sqrt{N}$. And then the bound on the result follows from Lemma 8 and the fact that $N_1 \leq N$. This also proves that the calls to **InertGenToldealOdd** and **InertGenToldealPowTwo** have a working bound of N^2 as the input is reduced mod $N_1/2^e\mathcal{O}_0$ and $2^e\mathcal{O}_0$ with $2^{2e} \leq N_1^2 \leq N^2$.

Finally, as we can assume that $a, b, c, d \leq N$, the Gaussian GCD computations have a working bound of N^2 , and the coefficients of the multiplication $\gamma(u - iv)$ is smaller than $2N^{3/2}$. This concludes the proof that the working bound to compute x, y is N^2 . $\square$

3.5 Intersection and multiplication, the generic case

Using the decomposition introduced in Lemma 7, we can also propose a generic algorithm for intersection of ideals of coprime norms that reduces to our inert ideal intersection algorithm. The algorithm is mainly based on the following lemma:

Lemma 14. *Let $I = I_1\alpha_1$ and $J = I_2\alpha_2$ be two ideals of coprime norm where I_1 and I_2 are in inert form, and $\alpha_1, \alpha_2 \in \mathbb{Z}[i]$. Let γ_1 be a generator of I_1 and γ_2 be a generator of I_2 , then $I \cap J = (K_1 \cap K_2)\alpha_1\alpha_2$ where K_1, K_2 are inert ideals of respective norm N_1, N_2 generated respectively by $\gamma_1\overline{\alpha_2}$ and $\gamma_2\overline{\alpha_1}$.*

Proof. First, since I and J have coprime norm, the norm of α_1 that divides $n(I_1)$ is coprime to $n(I_2)$ and $n(\alpha_2)$ is coprime to $n(I_1)$. Thus, it is clear that $\gamma_1\overline{\alpha_2}$ generates an ideal of norm $n(I_1)$ and $\gamma_2\overline{\alpha_1}$ generates an ideal of norm $n(I_2)$. Since I_1 is in inert form it contains an element of the form $(j + x + iy)/2$ that we can assume to be γ_1 without loss of generality. Now since $\alpha_2 \in \mathbb{Z}[i]$, we can see that $(j + x + iy)\overline{\alpha_2}/2 = (\alpha_2 j + \overline{\alpha_2}(x + iy))/2$. By multiplying on the left by $\overline{\alpha_2}$, and multiplying by an inverse of $n(\alpha_2) \pmod{n(J)}$ and reducing the result

mod $\mathfrak{n}(J) \mathcal{O}_0$, we obtain an element of the form $(j + x' + iy')/2$ that is contained in I_2 , and this proves that K_1 is in inert form. The same reasoning applies to K_2 .

Thus, K_1, K_2 are inert ideals of coprime norm. Their intersection is well-defined and has norm $\mathfrak{n}(I_1) \cdot \mathfrak{n}(I_2)$. This proves that $K = (K_1 \cap K_2) \alpha_1 \alpha_2$ has the same norm as $I \cap J$. It remains to show that it is contained in $I \cap J$ to prove the result.

We have $(K_1 \cap K_2) \cdot \alpha_1 \alpha_2 \subset K_1 \cdot \alpha_1 \alpha_2$. Furthermore, $K_1 \alpha_2 \subset I_1$ since $K_1 = \mathcal{O}_0 \gamma_1 \overline{\alpha_2} + \mathcal{O}_0 \mathfrak{n}(I_1)$, which proves $K_1 \alpha_2 = \mathcal{O}_0 \mathfrak{n}(\alpha_2) \gamma_1 + \mathcal{O}_0 \alpha_2 \mathfrak{n}(I_1) \subset \mathcal{O}_0 \gamma_1 + \mathcal{O}_0 \mathfrak{n}(I_1) = I_1$. Hence $K_1 \alpha_2 \alpha_1 \subset I_1 \alpha_1 = I$ which proves that $K \subset I$. The same reasoning applied symmetrically shows that K is contained in J . $\square$

Algorithm 5 Intersect

Input: $N_1, (x_1, y_1), \alpha_1, N_2, (x_2, y_2), \alpha_2$, such that $\text{GCD}(N_1 \mathfrak{n}(\alpha_1), N_2 \mathfrak{n}(\alpha_2)) = 1$ and

$I_1 = N_1, (x_1, y_1)$ and $I_2 = N_2, (x_2, y_2)$ are two inert ideals of coprime norms N_1, N_2

Output: $N, (X, Y), \alpha$ such that $(I_1 \cdot \alpha_1) \cap (I_2 \cdot \alpha_2) = K \cdot \alpha$ where K is the inert ideal $N, (X, Y)$

```

1:  $t_2 \leftarrow \mathfrak{n}(\alpha_2)^{-1} \pmod{N_1}$ 
2:  $X_1 + iY_1 \leftarrow (x_1 + i \frac{2y_1+1}{2}) \overline{\alpha_2}^2 \pmod{N_1 \mathbb{Z}[i]}$ 
3:  $Y_1 \leftarrow Y_1 - \frac{\mathfrak{n}(\alpha_2)}{2}$  //  $Y_1$  should be an integer
4:  $X_1 \leftarrow (X_1 t_2) \pmod{N_1}$ 
5:  $Y_1 \leftarrow (Y_1 t_2) \pmod{N_1}$ 
6:  $t_1 \leftarrow \mathfrak{n}(\alpha_1)^{-1} \pmod{N_2}$ 
7:  $X_2 + iY_2 \leftarrow (x_2 + i \frac{2y_2+1}{2}) \overline{\alpha_1}^2 \pmod{N_2 \mathbb{Z}[i]}$ 
8:  $Y_2 \leftarrow Y_2 - \frac{\mathfrak{n}(\alpha_1)}{2}$  //  $Y_2$  should be an integer
9:  $X_2 \leftarrow (X_2 t_1) \pmod{N_2}$ 
10:  $Y_2 \leftarrow (Y_2 t_1) \pmod{N_2}$ 
11:  $X \leftarrow \text{CRT}([X_1, X_2], [N_1, N_2])$ 
12:  $Y \leftarrow \text{CRT}([Y_1, Y_2], [N_1, N_2])$ 
13: Return  $(N_1 N_2, (X, Y), \alpha_1 \alpha_2)$ 

```

Lemma 15. *The output of Algorithm 5 is bounded by $N_1 N_2$ and the working bound is $2 \max(N_1, N_2)^2$.*

Proof. All the computations modulo N_1, N_2 can be done respectively with a working bound of N_1^2, N_2^2 , then the result follows from Lemma 27 which gives a working bound of $2 \max(N_1, N_2)^2$ on the call to $\text{CRT} \pmod{N_1, N_2}$. $\square$

Generic multiplication of $\mathcal{O}_0$ -ideals. The multiplication of generic $\mathcal{O}_0$ -ideals of coprime norms can be trivially reduced to the multiplication of inert ideals as $(I_1 \alpha_1) \cdot I_2 \alpha_2 = \overline{\alpha_1} (\overline{I_1} \cdot I_2) \alpha_2$. This is not an operation required for our application to SQIsign.

4 Application to the Deuring correspondence

In this section, we describe how our new algorithms from Section 3 can be integrated into higher-level building blocks to deal with ideal classes and sampling of ideals.

4.1 Ideal reductions

Reduction of ideals, in part to find short representatives of some ideal class, is an important operation for the Deuring correspondence, as it prevents a blow-up of the norm and degree of the ideals and isogenies involved.

Note that the latest update of the SQIsign introduced heavy changes to the lattice reduction part of the scheme following some yet unpublished work by Espitau, Schaeffler, van Woerden and Wallet. To avoid going too much into the intricacies of the new method, which is significantly more involved, we assume that the old method from the previous round [2] is still used. Concretely it consists in computing the Gram matrix of the ideal and applying the L2 algorithm [28]. Henceforth, we use the `IdealReduction` notation for this algorithm that we do not specify in full details. The bound we derive for this method applies to the new reduction method as well.

In Section A.2, we introduce optimized algorithms to compute the Gram matrix of inert ideals, and the multiplication of inert ideals. These are the only cases required for SQIsign. We also provide with Lemmas 29 and 30 a bound on the size of the Gram matrix, and on the working bound required to compute the Gram matrix with these algorithms. Together with Lemma 28, this allows us to derive the bounds we need for the rest of the paper.

4.2 Equivalent ideal computations

To reduce the norm of the ideals manipulated throughout the protocol, it is important to choose representatives of small norm within our ideal classes. Moreover, to ensure the coprimality conditions required by our multiplication and intersection algorithms, we also require some additional constraints on the norm of the output such as primality or coprimality to some value. This gives the `EquivalentCoprimeInertIdeal` algorithm that we describe below. To cover both prime and coprime case at once, we use the convention that $\text{GCD}(N, 0) = 1$ is equivalent to N being prime.

As explained at the end of Section 2.1, finding an equivalent ideal of given norm reduces to finding an element of given norm in the ideal. For that, common practice is to sample small random combinations of the reduced basis of the ideal until a suitable solution is found. The coefficients of this combination are taken from a box $[-b, b]$ for some small parameter b (this is the `ElBox` parameter from [1], and it is chosen to ensure success of the algorithm with overwhelming probability).

Moreover, for our purpose of bounding integers, it is important to bound the norm of the output of this algorithm. To do so, we assume that our algorithm only succeeds on input for which the fourth minimum is bounded by some value

B . Later in Section 5.3, we discuss the impact of choosing an “absolute” B that applies to all ideal classes, or a “probable” bound that applies only to an overwhelming proportion of all ideals.

Algorithm 6 EquivalentCoprimeInertIdeal

Input: an $\mathcal{O}_0$ -ideal I of norm N , an even integer M

Output: $\perp$, or an ideal $J \sim I$ of norm coprime to N in inert form, the element $\beta \in I$ such that $J = \chi_I(\beta)$

```

1:  $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \leftarrow \text{IdealReduction}(I)$  // Reduced basis
2: if  $\mathfrak{n}(\alpha_4)/N > B$  then
3:   return  $\perp$ 
4:  $N' \leftarrow M$ 
5: while true do
6:   Sample  $(c_1, c_2, c_3, c_4)$  uniformly at random from  $[-b; b]$ .
7:    $\beta \leftarrow \sum_{i=1}^4 c_i \alpha_i$ 
8:    $\beta \leftarrow \text{MakePrimitive}(\beta)$ 
9:    $N' \leftarrow \mathfrak{n}(\beta)/\mathfrak{n}(I)$ 
10:  if  $\text{GCD}(N', M) = 1$  then // Use convention that  $\text{GCD}(0, N') = 1$  iff  $N'$  prime
11:     $\theta, J \leftarrow \text{GenTolIdeal}(N', \beta)$ 
12:     $\beta \leftarrow \theta\beta/\mathfrak{n}(\theta)$ 
13:  return  $J, \beta$ 
```

Lemma 16. *The output of Algorithm 6 is bounded by b^2B , and the working bound is $\text{WB}(N)^2 + (b^2B)^2$.*

Proof. According to Lemma 29, the Gram matrix of I can be computed with a working bound of $p + 8N^2$, and $\|G\|_\infty \leq \text{WB}(N)$. Then, according to Lemma 28, the lattice reduction can be performed with a working bound of $\|G\|_\infty^2$ and the resulting basis have a gram matrix upper-bounded by p , and after the rejection step on the fourth minima, the upper-bound becomes B . The computation of β and its norm can be done with integers smaller than b^2B , and then, we conclude the proof with Lemma 8. $\square$

Additionally, some ideal-to-isogeny algorithms require to compute the shortest equivalent ideal. We provide pseudocode and bonds for this in Section A.3.

4.3 Ideal sampling methods

For cryptographic applications of the Deuring correspondence, it is useful to have algorithms to sample uniformly random ideals of a given norm: one for prime norm, and one for arbitrary norm. We make the distinction because the prime norm case is much more efficient (as the factorization is then trivial).

Sampling inert ideals of prime norm. Given the characterization of inert ideals given by Lemma 6, we get an algorithm to generate a random inert ideal of prime norm that are used for the secret key and commitment generation. Note that in the prime norm case, sampling only inert ideals is not a problem at all. First as there are only 2 out of $N + 1$ non-inert ideals of prime norm so the difference in distribution is quite negligible when N is of cryptographic size. Second, as shown in [25], non-inert ideals of prime norm are principal, and would thus lead to trivially weak public key or commitment curves. Thus avoiding non inert ideals is actually quite desirable, even though principal ideals are negligibly rare, so that not checking for them (as in previous specifications of SQIsign) causes no issues.

The idea is to sample x, y directly. Note that Algorithm 7 is restricted to prime norms because this is sufficient for sampling any ideal class. The exact same algorithm works for any norm of known factorization, as the only step of this algorithm which requires primality or a known factorization is the computation of a square root modulo the norm.

Algorithm 7 PrimeNormIdealInertSampling

Input: a prime odd norm $N \in \mathbb{N}$
Output: a uniformly random inert $\mathcal{O}_0$ -ideal of norm N

- 1: Sample a uniformly random $0 \leq x < N$.
- 2: **if** $-4x^2 - p$ is a square mod $4N$ **then**
- 3: $Y \leftarrow \sqrt{-4x^2 - p} \mod 4N$
- 4: Sample a uniformly random $b \in \{0, 1\}$
- 5: $Y \leftarrow (-1)^b Y \mod 2N$
- 6: $y \leftarrow (Y - 1)/2$
- 7: **else**
- 8: Go back to Step 1.
- 9: **return** $N, (x, y)$

Lemma 17. *If the modular square root algorithm used in Algorithm 7 is deterministic, then the output distribution is uniformly random among the set of inert ideals of norm N . The output is bounded by N , and the working bound is $p + 4N^2$.*

Proof. By Lemma 6, it is enough to sample a uniformly random element of the form $x + iy + (i + j)/2$ such that $0 \leq x, y < N$ and $n(x + iy + (i + j)/2) = x + iy + (i + j)/2 \mod N$. The choice of x is uniform, so if the modular squareroot algorithm is deterministic, the choice of b ensures that each of the two possible associated y is chosen with the same probability.

The bound is clear from the outline of the algorithm □

Sampling uniformly random ideal class. It is possible to sample an ideal class statistically close to the uniform distribution with PrimeNormIdealInertSampling

by taking a large enough prime input norm D_{mix} . For a statistical distance of $2^{-\lambda}$, [1] uses D_{mix} as the smallest prime above $p + 2^{2\lambda}$. For that reason D_{mix} is quite large, and so we use **PrimeNormIdeallnertSampling** in conjunction to **EquivalentCoprimeInertIdeal** to get a much smaller representative of the class. This yields the **UniformO0ClassSampling** algorithm. Due to the requirement of our application of this algorithm to SQIsign, we require this smaller representative to be odd, but this is somewhat arbitrary. Without any external constraint, the best choice would have been to take the smallest possible ideal in the class (which must be inert by Lemma 7).

Algorithm 8 UniformO0ClassSampling

Input: M an integer

Output: An ideal I where the class of I is sampled with small bias among the $\mathcal{O}_0$ -ideal classes different from the class of $\mathcal{O}_0$, and $n(I)$ is coprime to M if $M \neq 0$ and prime otherwise

- 1: $I \leftarrow \text{PrimeNormIdeallnertSampling}(D_{\text{mix}})$
 - 2: $I, \_ \leftarrow \text{EquivalentCoprimeInertIdeal}(I, M)$
 - 3: **return** I
-

Lemma 18. *The output of Algorithm 8 is bounded by b^2B and the working bound is $\text{WB}(D_{\text{mix}})^2 + (b^2B)^2$.*

Proof. For the working bound, we combine Lemma 17 and Lemma 16 to bound the working bounds of **PrimeNormIdeallnertSampling** and **EquivalentCoprimeInertIdeal** on their respective inputs and take the maximum. This results in $\max(p + 4D_{\text{mix}}^2, \text{WB}(D_{\text{mix}})^2 + (b^2B)^2)$. We can assume that $B > \frac{1}{2}\sqrt{p}$ as otherwise, by Minkowski's second theorem, we would have to every every single ideal. Thus $(b^2B)^2 > \frac{1}{2}p$ for any choice of $b > 1$, and thus $p + 4D_{\text{mix}}^2 < \text{WB}(D_{\text{mix}})^2 + (b^2B)^2$.

The output bound only depends on the one of **EquivalentCoprimeInertIdeal**. □

Sampling Ideals of given arbitrary norm. Finally, we introduce an algorithm to sample a random ideal of arbitrary norm. This one is used for sampling the auxiliary isogeny which needs to be uniformly random. Most importantly, it takes a norm for the output ideal. It also takes a quaternion α whose norm must be coprime to N . This requirement actually comes from the actual application of this algorithm to SQIsign, but note however that algorithm is correct for any α satisfying this requirement.

This algorithm also uses an implicit parameter m to ensure termination of **FullRepresentInteger**: the input mN to **FullRepresentInteger** needs to be big enough. We also require m to be coprime to N to ensure that γ generates an ideal of norm N , and that is why m is chosen as a prime number.

Algorithm 9 RandomIdealGivenNorm(N)

Input: An odd positive integer N coprime to p . A quaternion element α .

Output: A random left $\mathcal{O}_0$ -ideal J of norm N , or raise an exception.

```
1: found  $\leftarrow$  false
2:  $\gamma \leftarrow \text{FullRepresentInteger}(Nm, i, \mathcal{O}_0, \text{false})$ 
3: while not found do
4:    $x, y, z, w$  uniformly randomly selected integers in  $[1, N]$ 
5:    $\beta \leftarrow x + yi + zj + wj$ 
6:    $M \leftarrow \mathbf{n}(\beta) \bmod N$ 
7:   found  $\leftarrow (\text{GCD}(M, N) = 1)$ 
8:  $\gamma' \leftarrow \gamma\beta \bmod N\mathcal{O}_0$ 
9:  $\gamma' \leftarrow \gamma'\alpha \bmod N\mathcal{O}_0$ 
10:  $\alpha, J \leftarrow \text{GenTolIdeal}(\gamma', N)$ 
11: return  $\alpha, J$ 
```

Lemma 19. *The output of Algorithm 9 is bounded by N and the working bound is $N^2 + Nm$.*

Proof. It can be shown easily that the working bound of FullRepresentInteger is Nm , and the coefficients of γ are smaller than $\sqrt{Nm}$. Then, all the remaining computations $\bmod N$ can be done with a working bound of N^2 . The proof is concluded by Lemma 13. $\square$

4.4 Finding a representative of any ideal class

The building blocks introduced above give a way to represent every ideal class of maximal orders in $B_{p,\infty}$. Indeed, for any maximal orders $\mathcal{O}_1, \mathcal{O}_2$ in $B_{p,\infty}$ for which we know a connecting ideal to $\mathcal{O}_0$, we can compute I_1, I_2 two inert $\mathcal{O}_0$ -ideals of coprime norm whose right order is respectively isomorphic to $\mathcal{O}_1$ and $\mathcal{O}_2$. Then, our InertIdealMultiplication obtains $\overline{I}_1 \cdot I_2$, a representative of the ideal class whose left order is isomorphic to $\mathcal{O}_1$ and right order isomorphic to $\mathcal{O}_2$.

Note that this is the idea behind SQIsign's response computation which computes a small isogeny between two given supersingular curves i.e. a small norm element in an arbitrary ideal class. The multiplication doesn't directly give a small element, but once a concrete representative of the class has been computed, a small norm representative can be deduced from an ideal reduction step.

For translating this non- $\mathcal{O}_0$ -ideal to an isogeny, modern algorithms then compute the product of this short arbitrary ideal with an $\mathcal{O}_0$ -ideal I' to its left order of coprime norm (using a product of generators and norms), since only those can be directly translated to isogenies. The isogeny factor corresponding to I' is factored out on the isogeny side afterwards.

5 Application to SQIsign

In this section, we describe and analyze how to integrate our new algorithms to SQIsign. The algorithms are extracted from the recent updated specification

document [1]. We try to focus on the relevant building blocks, and refer the reader to the specification for a precise description of the parts of the scheme omitted in this document. In particular, we sometimes make some simplifications the sake of readability when this is not impacting our analysis. As verification is not using any quaternions, we focus on key generation and signing.

Our goal in this section is twofold: first, illustrate how the new algorithms are sufficient to cover all the quaternion computations required in SQIsign, second to bound the size of the integers involved in the quaternion operations throughout the whole key generation and signature process.

Some additional parameters of SQIsign. We introduce two parameters of the scheme that are going to be important for bounding the integers involved in the signature process:

1. The exponent f is such that $p = c2^f - 1$ where c is odd.
2. The response degree bound exponent e_{rsp} on the response isogeny.
3. The challenge degree exponent e_{chl} such that $D_{\text{chl}} = 2^{e_{\text{chl}}}$.

5.1 Building blocks

In this section, we introduce various algorithmic building blocks specific to SQIsign.

Sampling of a random bounded element in quaternion ideals. One of the most crucial step in SQIsign’s response is the computation of the quaternion elements that defines the response isogeny. In the latest version of the scheme, the degree of the response isogeny must be odd and smaller than $D_{\text{rsp}} = 2^{e_{\text{rsp}} - 1}$. Through the Deuring correspondence, this is equivalent to finding an element of norm qN such that q is odd and $q \leq D_{\text{rsp}}$, and N is the norm of some ideal (that we are going to describe in the next section). This is done with a `RandomBoundedElement` algorithm that we do not fully analyze for reasons detailed in Section 4.1. The main idea behind this algorithm is to use a basis of the ideal derived from a reduced basis of the dual lattice, and then sample the response quaternion a linear combination (with carefully bounded coefficients) of this basis. This is almost the algorithm used in the second round submission [2]. The difference in [1] is to scale the dual lattice by $4pN^2$ where N is the norm of the ideal to simplify the computation, and use a different lattice reduction algorithm. The resulting scaled dual lattice has the same discriminant as the original lattice.

While far from obvious, we are going to assume that the working bound for the reduction of the scaled dual lattice is the same as the working bound for the reduction of the ideal which is given by Lemma 30 and Lemma 28 as $\text{WB}(N)^2$. We state this as Assumption 1 below. This assumption is experimentally verified for the algorithms of [1]. Then, the sampling part finds an element of norm less than NB by sampling elements of the ideal of norm which are at worst slightly above ND_{rsp} . Since in SQIsign, $D_{\text{rsp}} \ll p$ and we consider that $N > p$, we can assume

that the working bound of this part of the algorithm is dominated by the working bound of the lattice reduction part. This justifies the following assumption that we are going to use to derive our final bound for the whole scheme. We define $B_{rbe}(N)$ to be an upper-bound on the working bound of **RandomBoundedElement** on input an ideal of norm N .

Assumption 1. *In $SQIsign$, we have $B_{rbe}(N) = \text{WB}(N)^2$.*

Sampling $SQIsign$'s response. The ideal on which **RandomBoundedElement** is called is constructed in the **SampleQuaternionResponse** algorithm below. It must be equivalent to $\overline{I_{sk,chl}} \cdot I_{com}$. To ensure coprimality of the norm to ensure that we can apply our fast multiplication and keep the integer bound as low as possible, we can replace $I_{sk,chl}$ by some equivalent $I'_{sk,chl}$ of prime norm. It will be coprime to $n(I_{com})$ with overwhelming probability as justified in [1, Chapter 9].

Algorithm 10 SampleQuaternionResponse

Input: $I_{sk,chl}, I_{com}$ two inert ideals

Output: $I'_{sk,chl}$ inert ideal equivalent to $I_{sk,chl}$ but of prime norm, α_{rsp} an element of the lattice $\Lambda = \overline{I'_{sk,chl}} \cdot I_{com}$, $\alpha_{sk,chl}$ an element of $I'_{sk,chl}$ and q_{rsp} equal to $n(\alpha_{rsp}) / (n(I_{com}) \cdot n(I'_{sk,chl}))$

- 1: $I'_{sk,chl}, \alpha_{sk,chl} \leftarrow \text{EquivalentCoprimeInertIdeal}(I_{sk,chl}, 0)$
 - 2: $\alpha_{sk,chl} \leftarrow \overline{\alpha_{sk,chl}}$
 - 3: $q_{rsp}, \alpha_{rsp} \leftarrow \text{RandomBoundedElement}(\text{InertIdealMultiplication}(I'_{sk,chl}, I_{com}), D_{rsp})$
 - 4: **return** $I'_{sk,chl}, \alpha_{rsp}, \alpha_{sk,chl}, q_{rsp}$
-

Lemma 20. *Let us write $C = b^2 B$. Assume that $n(I_{com}) \leq C$, and that $n(I_{sk,chl}) \leq D_{chl}C$, and let D_{rsp} be the upper-bound on the norm of the response, and B_{rbe} the working bound on **RandomBoundedElement**. Then, the working bound of **SampleQuaternionResponse** is*

$$\max(\text{WB}(CD_{chl})^2 + 4C^2, B_{rbe})$$

$$\text{and } n(I'_{sk,chl}) \leq C, n(\alpha_{sk,chl}) \leq C^2 D_{chl} \text{ and } n(\alpha_{rsp}) \leq C^2 D_{rsp}$$

Proof. From Lemma 16, we get that the first call to **EquivalentCoprimeInertIdeal** produces an output smaller than C with a working bound of

$$\text{WB}(CD_{chl})^2 + C^2.$$

Then, the computation of the multiplication with I_{com} gives an ideal of norm bounded by C^2 , represented by integers smaller than $2C^2$, and the working bound to make that computation is $4C^2$ by Lemma 12.

This proves the result on the working bound of the algorithm. □

Ideal to isogeny. `IdealTolsogeny` is the algorithm that takes an $\mathcal{O}_0$ -ideal and outputs a representation of the corresponding isogeny under the Deuring correspondence. A precise description of this algorithm is out of the scope of this work. In `SQIsign` [1], this is done with the `Qlapoty` variant [17] of the `Qlapoti` algorithm from [9]. The idea of this algorithm is: given an ideal I in input, find I_1, I_2 equivalent to I such that $n(I_1) + n(I_2) = 2^e$ for some $2^e \leq p$. For that, it first replaces I by the smallest element of its class (see Section A.3). For that reason, we are going to assume that once this reduction is performed, all integers manipulated in the ideal to isogeny algorithm are smaller than p^2 . With that we get the following lemma.

Lemma 21. *Let us write $C = b^2B$. Let I be an $\mathcal{O}_0$ -ideal of norm N , then the working bound of `IdealTolsogeny` on input I is $\max(p^2, \text{WB}(N)^2 + C^2)$.*

Proof. The bound of `SmallestEquivalentIdeal` is given by Lemma 31. The proof is completed our heuristic working bound of p^2 for the rest of the `IdealTolsogeny`.

Computation of the HD response. The computation of the HD response is one of the most involved and crucial step in the signature part of `SQIsign`. It only involves a few quaternion operations at the beginning to generate the auxiliary isogeny required by the HD response. We only focus on these steps, which are `framed` to highlight them. The execution of `Intersect` on $I_{\text{sk},\text{chl}}$ and I_{aux} is possible only if their norm are coprime to one another which will be the case in `SQIsign` with overwhelming probability as $I_{\text{sk},\text{chl}}$ will have prime norm.

Algorithm 11 HDResponse

Input: $I_{\text{sk},\text{chl}}$ an inert ideal, α_{rsp} an element of $\overline{I_{\text{sk},\text{chl}}}$ with $n(\alpha_{\text{rsp}}) < 2^{e_{\text{rsp}}-1}$, $\alpha_{\text{sk},\text{chl}}$ element of $I_{\text{sk},\text{chl}}$, and q_{rsp} an integer. The curve E_{com} and a basis $\mathcal{B}_{\text{com}}$ of $E_{\text{com}}[2^f]$

Output: A curve $E_{\text{aux}}, \mathcal{B}_{\text{aux}}$ a basis of $E_{\text{aux}}[2^{e_{\text{rsp}}+2}]$, a curve E_{chl} , and $\mathcal{B}_{\text{chl}}$ a basis of $E_{\text{chl}}[2^{e_{\text{rsp}}+2}]$

- 1:
- 2: $\theta_{\text{aux}}, I_{\text{aux}} \leftarrow \text{RandomIdealGivenNorm}(2^{e_{\text{rsp}}} - q_{\text{rsp}}, \alpha_{\text{sk},\text{chl}})$
- 3: $E'_{\text{aux}}, \mathcal{B}'_{\text{aux}} \leftarrow \text{IdealTolsogeny}(\text{Intersect}((\theta_{\text{aux}}, I_{\text{aux}}), I_{\text{sk},\text{chl}}))$
- 4: $\mathbf{M}_{\text{rsp}} \leftarrow \text{EndomorphismToMatrix}(\alpha_{\text{rsp}})$
- 5: $\mathbf{M}_{\text{aux}} \leftarrow \text{EndomorphismToMatrix}(\theta_{\text{aux}})$
- 6: $\mu \leftarrow \text{ModularInverse}(n(I_{\text{sk},\text{chl}}), 2^f)$
- 7: $\mathcal{B}'_{\text{aux}} \leftarrow \text{ActOnBasis}(\mathcal{B}'_{\text{aux}}, \mu \cdot \mathbf{M}_{\text{aux}} \mathbf{M}_{\text{rsp}})$
- 8: $q_{\text{inv}} \leftarrow \text{ModularInverse}(q_{\text{rsp}}, 2^{e_{\text{rsp}}+2})$
- 9: $(P_{\text{com}}, Q_{\text{com}}) \leftarrow [2^{f-e_{\text{rsp}}-2}] \mathcal{B}_{\text{com}}$
- 10: $(P'_{\text{aux}}, Q'_{\text{aux}}) \leftarrow [q_{\text{inv}} 2^{f-e_{\text{rsp}}-2}] \mathcal{B}'_{\text{aux}}$
- 11: $E_{\text{aux}}, E_{\text{chl}}, [\mathcal{B}_{\text{aux}}, \mathcal{B}_{\text{chl}}], \_$
- 12: $\leftarrow \text{DimTwoKaniIsogeny}(E_{\text{com}}, (P_{\text{com}}, Q_{\text{com}}), E'_{\text{aux}}, (P'_{\text{aux}}, Q'_{\text{aux}}), e_{\text{rsp}}, [(P_{\text{com}}, 0_{E_{\text{com}}}), (Q_{\text{com}}, 0_{E'_{\text{aux}}})], \text{false})$
- 13: **return** $E_{\text{aux}}, \mathcal{B}_{\text{aux}}, E_{\text{chl}}, \mathcal{B}_{\text{chl}}$

Lemma 22. Assume $n(I_{sk,chl}) \leq C$, then the working bound of **HDResponse** is

$$2^{e_{rsp}}m + 2 \max(2^{e_{rsp}}, C, p, \text{WB}(C2^{e_{rsp}}))^2.$$

5.2 SQIsign keygen and signature algorithms

Below we give a simplified description of the keygen and signature algorithms from [1]. The important steps for bounding the quaternion part are framed. We deduce results on the maximum size of the integers involved from all the results proven throughout the last few sections.

Algorithm 12 KeyGen(sk, pk)

Output: sk, pk a SQIsign key pair.

- 1: $I_{sk} \leftarrow \text{UniformO0ClassSampling}(0)$
 - 2: $E_{pk}, \varphi_{sk}(\mathcal{B}_0) \leftarrow \text{IdealTolsogeny}(I_{sk})$
 - 3: $\mathcal{B}_{pk}, \text{hint}_{pk} \leftarrow \text{TorsionBasisFromHint}(E_{pk}, e_{chl} + 2, 0)$
 - 4: $\mathbf{M}_{sk} \leftarrow \text{ChangeOfBasis}(E_{pk}, e_{chl} + 2, \varphi_{sk}(\mathcal{B}_0), \mathcal{B}_{pk})$
 - 5: $\mathbf{M}_{sk} \leftarrow \mathbf{M}_{sk} \bmod 2^{e_{chl}}$
 - 6: $\mathbf{M}_{sk} \leftarrow \text{NormalizeMatrix}(\mathbf{M}_{sk}, e_{chl})$
 - 7: $\text{pk} \leftarrow (E_{pk}, \text{hint}_{pk})$
 - 8: $\text{sk} \leftarrow (\text{pk}, I_{sk}, \mathbf{M}_{sk})$
 - 9:
 - 10: **return** sk, pk
-

Proposition 23. Let $C = b^2B$. The working bound of **KeyGen** is

$$2 \max(\text{WB}(D_{mix}), \text{WB}(C), C, p)^2,$$

and $n(I_{sk}) \leq C$.

Proposition 24. Let $C = b^2B$. The working bound of **Sign** is :

$$2^{e_{rsp}}m + 4 \max\left(2^{e_{rsp}}, C, p, \text{WB}(C2^{e_{rsp}}), \text{WB}(CD_{chl}), \text{WB}(D_{mix}), \sqrt{B_{rbe}}\right)^2$$

Under Assumption 1, this gives the upper-bound to:

$$2^{e_{rsp}}m + 4 \max\left(2^{e_{rsp}}, C, p, \text{WB}(C2^{e_{rsp}}), \text{WB}(CD_{chl}), \text{WB}(D_{mix}), \text{WB}(C^2)\right)^2$$

5.3 Numerical Application to SQIsign's parameter sets.

In this section, we make the numerical application to give the precise bounds for concrete parameters sets of SQIsign. For completeness and a comparison with [18], we apply it to both the old SQIsign parameter from [2], and the new

Algorithm 13 Sign(sk, msg)

Input: Secret signing key sk, a message msg $\in \{0, 1\}^*$ **Output:** Signature σ

- 1: Parse sk as (pk, I_{sk}, M_{sk})
 - // **Commitment**
 - 2: $I_{com} \leftarrow \text{UniformO0ClassSampling}(2)$ // Sample a random $\mathcal{O}_0$ -ideal of odd norm
 - 3: $E_{com}, \mathcal{B}_{com} \leftarrow \text{IdealTolsogeny}(I_{com})$
 - // **Challenge**
 - 4: chl $\leftarrow H(pk, E_{com}, msg)$
 - // **Response**
 - 5: $(c_1, c_2) \leftarrow M_{sk} \cdot [1, chl]^T$
 - 6: $I_{chl} \leftarrow \text{KernelDecomposedToIdeal}(e_{chl}, c_1, c_2)$
 - 7: $I_{sk,chl} \leftarrow \text{Intersect}(I_{chl}, I_{sk})$ // $I_{sk,chl} = I_{chl} \cap I_{sk}$
 - 8: $I_{sk,chl}, \alpha_{rsp}, \alpha_{sk,chl}, q_{rsp} \leftarrow \text{SampleQuaternionResponse}(I_{sk,chl}, I_{com})$
 - // **Response packaging**
 - 9: $E_{aux}, \mathcal{B}_{aux}, E_{chl}, \mathcal{B}_{chl} \leftarrow \text{HDResponse}(I_{sk,chl}, \alpha_{rsp}, \alpha_{sk,chl}, q_{rsp}, E_{com}, \mathcal{B}_{com})$
 - 10: $M_{chl}, \text{hint}_{aux}, \text{hint}_{chl} \leftarrow \text{SetChangeOfBasisMatrix}(E_{aux}, E_{chl}, \mathcal{B}_{aux}, \mathcal{B}_{chl}, e_{rsp} + 2)$
 - 11: $M_{chl} \leftarrow \text{NormalizeMatrix}(M_{chl}, e_{rsp} + 2)$
 - 12: **return** $\sigma \leftarrow (chl, E_{aux}, M_{chl}, \text{hint}_{aux}, \text{hint}_{chl})$
-

SQIsign parameters from [1]. We remind the readers that the parameters had to be modified following [33]. We also assume that Assumption 1 holds, which has been experimentally verified in practice for all SQIsign parameter sets.

Most of the parameters in Propositions 23 and 24 have a formula depending mainly on the security level, and prime p . We provide special treatment for the bound B for which we consider two possible values. One *absolute* bound B_{abs} that applies to any ideal, and one *probable* bound B_{prob} that applies to an overwhelming proportion of all ideals. Here overwhelming is defined with respect to the security level λ , meaning that the probability that this bound is not satisfied is smaller than $2^{-\lambda}\mu$ for some small constant μ . More concretely, for B we use the bound $47p/\pi^2$ on the output of the L2 algorithm proven in [18, Lemma 3]. For B_{prob} , the idea is that for $\mathcal{O}_0$ -ideals, the first and second minimum, and third and fourth minimum are equal since $\mathcal{O}_0$ contains i of norm 1. Thus, Minkowski's second theorem gives $\lambda_1^2 \lambda_4^2 = \Theta(p^2)$. There are $O(M)$ left $\mathcal{O}_0$ -ideals of norm M , and so there are $O(M^2)$ ideals of norm smaller than some bound M . Thus, by fixing $M = p/B_{prob}$, the probability that we get an ideal whose fourth minimum is larger than B_{prob} is proportional to M^2/p . Thus, we take $M \approx \sqrt{p}2^{\lambda/2}$.

The values of the parameters (most of them are extracted from [1]) are summarized in Table 1. When the formula was not simple enough, we used the same parameter for both old and new parameter sets.

From the parameters, one sees that the two larger terms by far in the formula given in Proposition 24 are $4WB(D_{mix})^2$ and $4WB(C^2)^2$ for $C = b^2B$. Moreover,

NIST level	λ	p	D_{mix}	D_{chl}	D_{rsp}	B_{abs}	B_{prob}	b	m
I - old	128	251	507	128	158	254	190	15	147
I - new	128	326	582	128	195	329	227	15	147
II - old	192	382	767	192	239	385	288	19	221
II - new	192	505	889	192	301	508	349	19	221
III - old	256	505	1017	256	317	508	381	22	286
III - new	256	669	1181	256	399	672	463	22	286

Table 1: SQIsign parameters at all (old and new) security level (in bitsize)

when $x \gg p$ we make the approximation $\text{WB}(x) \approx 4x$. This is true for C^2 and D_{mix} . Thus, we simplify the formula from Proposition 24 to be:

$$64 \max(D_{\text{mix}}^2, C^4)$$

for our concrete computations, which are reported in Table 2. We express the result in terms of maximal required bitsize.

Work	Fourth minimum bound	I-old	I-new	II-old	II-new	III-old	III-new
Ours	B_{prob}	1019	1170	1539	1738	2040	2367
Ours	B_{abs}	1051	1352	1578	2069	2069	2722
[18]	B_{abs}	1665	—	2521	—	3319	—

Table 2: Bound on the maximum bitsize required to run a SQIsign key generation and signature.

Note that the NIST submission currently uses slightly higher bounds due to some integer and quaternion tests which either assume a bound of p^5 or contain hard-coded constants of that size. However, one can easily verify that our bounds apply to the signature code in itself. For this, one adapts the precomputation scripts of [1] accordingly (rounding up to the next full word), reruns the precomputations and compiles without the failing mp, prng and quaternion tests and without any debug code. All other tests pass, in particular full signature tests.

References

1. Aardal, M.A., Adj, G., Aranha, D.F., Basso, A., Borin, G., Canales Martínez, I.A., Chávez-Saab, J., Corte-Real Santos, M., Dartois, P., De Feo, L., Duparc, M., Espitau, T., Eriksen, J.K., Fouotsa, T.B., Gazzoni Filho, D.L., Hess, B., Invernizzi, R., Kohel, D., Leroux, A., Longa, P., Maino, L., Meyer, M., Mula, M., Nakagawa, K.,

- Onuki, H., Panny, L., Patranabis, S., Petit, C., Pope, G., Reijnders, K., Robert, D., Rodríguez-Henríquez, F., Schaeffler, S., Vercauteren, F., Wallet, A., Wesolowski, B., van Woerden, W.: SQIsign. Tech. rep., National Institute of Standards and Technology (2026), <https://sqisign.org>
2. Aardal, M.A., Adj, G., Aranha, D.F., Basso, A., Canales Martínez, I.A., Chávez-Saab, J., Corte-Real Santos, M., Dartois, P., De Feo, L., Duparc, M., Eriksen, J.K., Fouotsa, T.B., Gazzoni Filho, D.L., Hess, B., Kohel, D., Leroux, A., Longa, P., Maino, L., Meyer, 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 (2025), <https://sqisign.org>
 3. Amorós, L., Clements, J., Martindale, C.: Parametrizing maximal orders along supersingular ℓ -isogeny paths. Cryptology ePrint Archive, Paper 2025/033 (2025), <https://eprint.iacr.org/2025/033>
 4. 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: IACR International Conference on Public-Key Cryptography. pp. 300–332. Springer (2025)
 5. Basso, A., Borin, G., Castryck, W., Corte-Real Santos, M., Invernizzi, R., Leroux, A., Maino, L., Vercauteren, F., Wesolowski, B.: PRISM with a pinch of salt: Simple, efficient and strongly unforgeable signatures from isogenies (2026)
 6. Basso, A., Borin, G., Corte-Real Santos, M., Dartois, P., Invernizzi, R., Maino, L., Pedersen, R., Seck, M.: Isogeny-based signatures with randomizable keys (2026)
 7. 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: Advances in Cryptology – ASIACRYPT 2024: 30th International Conference on the Theory and Application of Cryptology and Information Security, Kolkata, India, December 9–13, 2024, Proceedings, Part III. p. 339–370. Springer-Verlag, Berlin, Heidelberg (2024). https://doi.org/10.1007/978-981-96-0891-1_11, https://doi.org/10.1007/978-981-96-0891-1_11
 8. Basso, A., He, C., Jacquemin, D., Kouider, F., Kutas, P., Mukherjee, A., Schaeffler, S., Roy, S.S.: Constant-time quaternion algorithms for SQIsign. Cryptology ePrint Archive, Report 2025/2192 (2025), <https://eprint.iacr.org/2025/2192>
 9. 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: International Conference on the Theory and Application of Cryptology and Information Security. pp. 174–205. Springer (2025)
 10. Borin, G., De Feo, L., Lido, G., Schaeffler, S.: The SQInstructor: a guide to SQIsign and the deuring correspondence with level structures. In: Annual International Cryptology Conference. pp. 537–569. Springer (2026)
 11. Borin, G., Lai, Y.F., Leroux, A.: Erebor and Durian: Full anonymous ring signatures from quaternions and isogenies. IACR Communications in Cryptology (2024)
 12. Clements, J.: Structural results for maximal quaternion orders and connecting ideals of prime power norm in $B_{p,\infty}$. Cryptology ePrint Archive (2025)
 13. Czaprynyko, M., Mukherjee, A., Sinha Roy, S.: Tighter and friendlier integer bounds for quaternion algorithms - application to SQIsign. Cryptology ePrint Archive, Paper 2026/2049 (2026), <https://eprint.iacr.org/2026/2049>
 14. Dartois, P., Leroux, A., Robert, D., Wesolowski, B.: SQIsignHD: new dimensions in cryptography. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 3–32. Springer (2024)

15. De Feo, L., Kohel, D., Leroux, A., Petit, C., Wesolowski, B.: SQISign: compact post-quantum signatures from quaternions and isogenies. In: *Advances in Cryptology—ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security*, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part I 26. pp. 64–93. Springer (2020)
16. Duparc, M., Fouotsa, T.B.: SQIPrime: A dimension 2 variant of SQISignHD with non-smooth challenge isogenies. In: *International Conference on the Theory and Application of Cryptology and Information Security*. pp. 396–429. Springer (2024)
17. Duparc, M., Leroux, A., Schaeffler, S.: Qlapoty: Improved analysis and efficiency for quaternionic ideal to isogeny transformation. *Cryptology ePrint Archive* (2026)
18. Kim, W., Lee, C., Yoo, H.: Compact quaternion algorithms for SQISign. *Cryptology ePrint Archive*, Paper 2026/1031 (2026), <https://eprint.iacr.org/2026/1031>
19. Kim, W., Lee, J., Kim, H., Lee, C.: SQISign with fixed-precision integer arithmetic. In: *IACR International Conference on Public-Key Cryptography*. pp. 3–30. Springer (2026)
20. Kohel, D.R., Lauter, K., Petit, C., Tignol, J.P.: On the quaternion ℓ -isogeny path problem. *LMS Journal of Computation and Mathematics* **17**(A), 418–432 (2014). <https://doi.org/10.1112/S1461157014000151>
21. Kouider, F., Mukherjee, A., Jacquemin, D., Kutas, P.: Constant-time integer arithmetic for SQISign. In: Nitaj, A., Petkova-Nikova, S., Rijmen, V. (eds.) *AFRICACRYPT 25*. LNCS, vol. 15651, pp. 192–215. Springer, Cham (July 2025). https://doi.org/10.1007/978-3-031-97260-7_10
22. Leroux, A.: Quaternion algebra and isogeny-based cryptography. Theses, Ecole doctorale de l’Institut Polytechnique de Paris (2022)
23. Leroux, A.: Computation of hilbert class polynomials and modular polynomials from supersingular elliptic curves. *arXiv preprint arXiv:2301.08531* (2023)
24. Leroux, A.: Verifiable random function from the deuring correspondence and higher dimensional isogenies. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. pp. 167–194. Springer (2025)
25. Leroux, A.: Efficient quaternion algorithms for the deuring correspondence, and application to the evaluation of modular polynomials. *ANTS XVII* (2026)
26. Leroux, A., Roméas, M.: Compact, efficient and cca-secure updatable encryption from isogenies. *Cryptology ePrint Archive* (2025)
27. 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: *International Conference on the Theory and Application of Cryptology and Information Security*. pp. 272–303. Springer (2024)
28. Nguyen, P.Q., Stehlé, D.: An LLL algorithm with quadratic complexity. *SIAM Journal on Computing* **39**(3), 874–903 (2009). <https://doi.org/10.1137/070705702>, <https://doi.org/10.1137/070705702>
29. Santos, M.C.R., Eriksen, J.K., Leroux, A., Meyer, M., Panny, L.: Evaluation of modular polynomials from supersingular elliptic curves. *Lucant* (2025)
30. Silverman, J.H.: *The Arithmetic of Elliptic Curves*, Graduate Texts in Mathematics, vol. 106. Springer-Verlag (1986)
31. team, G.: The GNU Multiple Precision Arithmetic Library. <https://gmplib.org> consulted 2026 (1991)
32. Voight, J.: *Quaternion algebras*, vol. 288. Springer (2021)
33. 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 Additional content

A.1 Bounds on some classical algorithms

This section introduces various classical building blocks on which we rely.

Integer GCD. We give below a classical result on the output and working space required by the extended GCD algorithm.

Lemma 25. *The output of the $XGCD(a, b)$ are bounded by $\max(a, b)$ and this is also the working bound for the algorithm.*

Gaussian GCD. We present here an algorithm to compute the GCD over the ring $\mathbb{Z}[i]$ of Gaussian integers. This algorithm is based on the euclidian division in $\mathbb{Z}[i]$ that we define below. We remind the reader that division, and the GCD are well defined in $\mathbb{Z}[i]$ up to multiplication by $\mathbb{Z}[i]^\times = \langle i \rangle$.

If $\alpha = a + ib$ and $\beta = c + id$ are two elements of $\mathbb{Z}[i]$, then the quotient κ of α by β in $\mathbb{Z}[i]$ is the rounding of $\alpha/\beta = \alpha\bar{\beta}/n(\beta)$ which is equal to:

$$\kappa = \left\lfloor \frac{ac + bd}{c^2 + d^2} \right\rfloor + i \left\lfloor \frac{bc - ad}{c^2 + d^2} \right\rfloor.$$

Then, the reminder ρ of the division is $\alpha - \kappa\beta$. We write $\kappa, \rho = \text{GaussDiv}(\alpha, \beta)$. The algorithm **GaussGCD** is then defined very similarly to the Euclid's algorithm from the euclidian division.

Algorithm 14 GaussGCD

Input: $\alpha, \beta \in \mathbb{Z}[i]$.

Output: $\text{GCD}_{\mathbb{Z}[i]}(\alpha, \beta)$

```

1:  $\kappa, \rho \leftarrow \alpha, \beta$ 
2: if  $n(\kappa) < n(\rho)$  then
3:    $\kappa, \rho \leftarrow \rho, \kappa$ 
4: if  $n(\rho) = 1$  then
5:   return  $\rho$ 
6: if  $\rho = 0$  then
7:   return  $\kappa$ 
8:  $\kappa, \rho \leftarrow \text{GaussDiv}(\kappa, \rho)$ 
9: return GaussGCD( $\kappa, \rho$ )

```

Lemma 26. *The output of Algorithm 14 is bounded by $\sqrt{\max(n(\alpha), n(\beta))}$, and the working bound is $\max(n(\alpha), n(\beta))$.*

Proof. Let us take $\kappa, \rho \leftarrow \text{GaussDiv}(\alpha, \beta)$. If we write $\gamma = \alpha/\beta$, then it is easy to see that by definition $n(\gamma - \kappa) \leq 1/2$. Thus

$$n(\kappa) \leq \frac{1}{2} + \frac{n(\alpha)}{n(\beta)}.$$

Thus, when $n(\beta) > 1$, we get that $n(\kappa) < n(\alpha)$. Moreover, since $\rho = \beta(\gamma - \kappa)$ one gets that $n(\rho) < n(\beta)$. This proves that the norm of the input the successive calls is decreasing which proves termination, but also that the norm of the output is smaller than that of the input and so the coefficients of the output are bounded by $\sqrt{\max(n(\alpha), n(\beta))}$. Remains to bound the computations involved in the gaussian division. Since the coefficients of α, β are also smaller than $\sqrt{\max(n(\alpha), n(\beta))}$. The numerator of the fractions defining γ are the coefficients of $\alpha\beta$ which are smaller than $\max(n(\alpha), n(\beta))$, and then computing κ only uses smaller integers. Similary, computing ρ requires mainly to compute $\beta\kappa$, whose coefficients can be bounded by the same bound. $\square$

Chinese Remainder Theorem. Let N_1, N_2 be two coprime integers, and let $0 \leq a_i < N_i$ for $1 \leq i \leq 2$. We define $\text{CRT}([a_1, a_2], [N_1, N_2])$ to be the element $0 \leq x < N_1, N_2$ such that $x = a_i \pmod{N_i}$ for $1 \leq i \leq 2$.

Lemma 27. *Let N_1, N_2 be two coprime integers, and let $0 \leq a_i < N_i$ for $1 \leq i \leq 2$. Then $0 \leq \text{CRT}([a_1, a_2], [N_1, N_2]) < N_1 N_2$ and the working bound is of an algorithm computing this value is $2 \max(N_1, N_2)^2$.*

Proof. We have $x = (a_1 u_2 N_2 + a_2 u_1 N_1) \pmod{N_1 N_2}$ where $u_1 N_1 + u_2 N_2 = 1$. The computation of u_1, u_2 can be done using integers smaller than $\max(N_1, N_2)$ and their respective values is smaller than the same bound. It is clear that by definition the result is smaller than $N_1 N_2$.

Then, we get the desired working bound with

$$x = (N_2((a_1 u_2) \pmod{N_1}) + N_1((a_2 u_1) \pmod{N_2})) \pmod{N_1 N_2}$$

$\square$

Lattice reduction. Lattice reduction is an important building block for the Deuring correspondence, in particular to choose efficiently

Lemma 28. *Let G be the gram matrix of basis for an $\mathcal{O}_0$ -ideal I . Then, the entries of the Gram matrix of a Minkowski reduced basis of I are upper-bounded by p , and there is an algorithm to compute it with working bound $\|G\|_\infty^2$.*

Proof. The bound on the norm of the output is a well-known fact and follows from the existence of a non-trivial automorphism in $\mathcal{O}_0$ which implies that the 3rd and 4th successive minimum have the same norm.

We take the working bound of $\|G\|_\infty^2$ from [18, Lemma 8] which is quite classical for lattice reduction for instance using the L2 algorithm [28]. $\square$

A.2 Gram matrix computation

The computation of the Gram matrix of ideals in the form $M_{\text{inert}}^N(X, Y)$ or $M_{\times, \text{inert}}^{N_1 N_2}(X_1, X_2, Y_1, Y_2)$ can also be done quite efficiently. We present below two algorithms to do that computation.

Algorithm 15 InertGramMatrix

Input: an integer N , and an inert ideal (x, y) of norm N

Output: The gram matrix of the basis $M_{\text{inert}}^N(x, y)$

```

1:  $G \leftarrow (0)_{1 \leq i, j \leq 4}$ 
2:  $G_{1,1} \leftarrow 2N$ 
3:  $G_{2,2} \leftarrow 2N$ 
4:  $G_{3,3} \leftarrow (p + 4x^2 + (2y + 1)^2)/(2N)$ 
5:  $G_{4,4} \leftarrow G_{3,3}$ 
6:  $G_{1,3} \leftarrow 2x$ 
7:  $G_{1,4} \leftarrow -2y - 1$ 
8:  $G_{2,3} \leftarrow 2y + 1$ 
9:  $G_{2,4} \leftarrow 2x$ 
10: Complete  $G$  so that  $G_{i,j} = G_{j,i}$  for all  $1 \leq i, j \leq 4$ 
11: return  $G$ 

```

Lemma 29. *Let G be the output of Algorithm 15 on input $I = (x, y)$ of norm N . Then, $\|G\|_\infty \leq \text{WB}(N)$, and the working bound of InertGramMatrix is $(p + 8N^2)$.*

Proof. The biggest integer involved in the computation of G is clearly $p + 4x^2 + (2y + 1)^2$, since $x, y < N$, we get that this integer is smaller than $p + 8N^2$. The bound on the maximum value of G follows directly from there as all the other values are smaller than $2N$.

Lemma 30. *Let G be the output of Algorithm 16 on input $I_1 = (x_1, y_1)$ of norm N_1 and $I_2 = (x_2, y_2)$ of norm N_2 . Then, with $N = N_1 N_2$, we have $\|G\|_\infty \leq \text{WB}(N)$, and the working bound of GramMatrixMult is $(p + 16N^2)$.*

Proof. The proof is quite similar to the proof of Lemma 29, with the additional facts that $|A, B, C| \leq 16N^2$, and that $|G_{3,3}, G_{3,4}| \leq 8N_1 N_2$.

A.3 Smallest equivalent ideal computation

Some ideal-to-isogeny algorithms such as Qlapoti [9] and its variant Qlapoty [17] require the computation of an equivalent ideal of minimal norm.

In most cases this is very straightforward: It suffices to compute the shortest element of the input ideal, and take an equivalence using it. However, complications occur if the norm of the shortest equivalent and the input ideal are not coprime to one another. In this case an intermediary ideal equivalent to the input of norm coprime to the desired output is computed (in particular to ensure that the generator that we use is N -primitive).

Algorithm 16 GramMatrixMult

Input: two integers N_1, N_2 , and four integers X_1, X_2, Y_1, Y_2

Output: The Gram matrix of the basis $M_{\times, \text{inert}}^N(X_1, X_2, Y_1, Y_2)$

```
1:  $G \leftarrow (0)_{1 \leq i, j \leq 4}$ 
2:  $G_{1,1} \leftarrow 2N_1N_2$ 
3:  $G_{2,2} \leftarrow 2N_1N_2$ 
4:  $G_{3,3} \leftarrow (p + 4(X_1 + X_2)^2 + (Y_2 - Y_1)^2)/(2N_1N_2)$ 
5:  $A \leftarrow Y_1Y_2$ 
6:  $B \leftarrow 4X_1X_2$ 
7:  $C \leftarrow (Y_1 + 2X_1)(Y_2 + 2X_2)$ 
8:  $G_{4,4} \leftarrow G_{3,3} + (A - B)/(N_1N_2)$       //  $G_{4,4} \leftarrow G_{3,3} + (Y_1Y_2 - 4X_1X_2)/(N_1N_2)$ 
9:  $G_{3,4} \leftarrow (-C + A + B)/(N_1N_2)$           //  $G_{3,4} \leftarrow -2(X_2Y_1 + X_1Y_2)/(N_1N_2)$ 
10:  $G_{1,3} \leftarrow 2(X_1 + X_2)$ 
11:  $G_{1,4} \leftarrow -(Y_1 + Y_2)$ 
12:  $G_{2,3} \leftarrow (Y_2 - Y_1)$ 
13:  $G_{2,4} \leftarrow 2(X_2 - X_1)$ 
14: Complete  $G$  so that  $G_{i,j} = G_{j,i}$  for all  $1 \leq i, j \leq 4$ 
15: return  $G$ 
```

Lemma 31. Denote N the norm of the input ideal.

The output of Algorithm 17 is bounded by $\sqrt{p}$.

The working bound is bounded by the working bound of **EquivalentCoprimeInertIdeal**, that is $\text{WB}(N)^2 + (b^2B)^2$.

Proof. Minkowski's bound implies than any ideal class contains an element of norm less than $\sqrt{p}$ which gives the bound on the ideal representation. Since α_1 is computed modulo the output ideal norm, its coefficients are also below that bound.

All operations in **SmallestEquivalentIdeal** are also used in **EquivalentCoprimeInertIdeal**, where the occurring numbers are larger because of the additional sampling bound b . Additionally, **SmallestEquivalentIdeal** calls **EquivalentCoprimeInertIdeal** on its own input, so its working bound is exactly the same as for **EquivalentCoprimeInertIdeal**. $\square$

B Delayed proofs

Proofs for bounds in Section 5.

Lemma 22. Assume $n(I_{sk,chl}) \leq C$, then the working bound of **HDResponse** is

$$2^{e_{rsp}}m + 2 \max(2^{e_{rsp}}, C, p, \text{WB}(C2^{e_{rsp}}))^2.$$

Proof. The working bound for **RandomIdealGivenNorm** is $2^{e_{rsp}}(2^{e_{rsp}} + m)$ by Lemma 19. Then, by Lemma 15, the working bound for the intersection is $2 \max(2^{e_{rsp}}, C)^2$. Finally, the working bound for **IdealTolsogeny** is $p^2 + \text{WB}(C2^{e_{rsp}})^2$

Algorithm 17 SmallestEquivalentIdeal

Input: I ideal in inert form

Output: J ideal in inert form and of minimal norm in the equivalence class of I

```

1: and  $\alpha_1 \in I$  such that  $J = I_{\frac{\overline{\alpha_1}}{n(I)}}$ 
2:  $N \leftarrow n(I)$ 
3:  $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \leftarrow \text{IdealReduction}(I)$     // Assume  $\alpha_1$  is now of minimal norm in  $I$ 
4:  $N' \leftarrow n(\alpha_1) / N$ 
5: if  $\text{GCD}(N, N') \neq 1$  then
6:    $K, \mu \leftarrow \text{EquivalentCoprimeInertIdeal}(I, 2N'N)$ 
7:    $\beta \leftarrow (\frac{1}{N}\mu\overline{\alpha_1}) \bmod N'\mathcal{O}_0$ 
8: else
9:    $\beta \leftarrow (\frac{1}{N}\overline{\alpha_1}) \bmod N'\mathcal{O}_0$ 
10:  $J \leftarrow \text{InertGenToIdealOdd}(\beta, N')$     // An ideal shortest in its class is always inert
11: return  $J, \alpha_1$ 

```

by Lemma 21. The final expression is obtained by simplifying a bit the expression and regrouping every term under within the max function. The factor 2 allows to provide an upper bound and simplifies further the formula.

Proposition 23. *Let $C = b^2B$. The working bound of **KeyGen** is*

$$2 \max(\text{WB}(D_{\text{mix}}), \text{WB}(C), C, p)^2,$$

and $n(I_{\text{sk}}) \leq C$.

Proof. This follows directly from Lemma 18 and Lemma 21 □

Proposition 24. *Let $C = b^2B$. The working bound of **Sign** is :*

$$2^{e_{\text{rsp}}}m + 4 \max\left(2^{e_{\text{rsp}}}, C, p, \text{WB}(C2^{e_{\text{rsp}}}), \text{WB}(CD_{\text{chl}}), \text{WB}(D_{\text{mix}}), \sqrt{B_{\text{rbe}}}\right)^2$$

Under Assumption 1, this gives the upper-bound to:

$$2^{e_{\text{rsp}}}m + 4 \max\left(2^{e_{\text{rsp}}}, C, p, \text{WB}(C2^{e_{\text{rsp}}}), \text{WB}(CD_{\text{chl}}), \text{WB}(D_{\text{mix}}), \text{WB}(C^2)\right)^2$$

Proof. The bound for the computation of the commitment isogeny is the same as the one for the key generation given by Proposition 23:

$$2 \max(\text{WB}(D_{\text{mix}}), \text{WB}(C), C, p)^2,$$

Then, by the same proposition, the norm of I_{sk} is smaller than C . The norm of the challenge ideal is $2^{e_{\text{chl}}} = D_{\text{chl}}$, so by Lemma 15, the intersection of the secret key and challenge ideal has a working bound of $2 \max(D_{\text{chl}}, C)^2$. Then with Lemma 20, the working bound of the call to **SampleQuaternionResponse** is

$$\max(\text{WB}(CD_{\text{chl}})^2 + 4C^2, B_{\text{rbe}}).$$

Finally, Lemma 22 gives a working bound of

$$2^{e_{\text{rsp}}}m + 2 \max(2^{e_{\text{rsp}}}, C, p, \text{WB}(C2^{e_{\text{rsp}}}))^2.$$

We upper-bound all this by

$$2^{e_{\text{rsp}}}m + 4 \max(2^{e_{\text{rsp}}}, C, p, \text{WB}(C2^{e_{\text{rsp}}}), \text{WB}(CD_{\text{chl}}), \text{WB}(D_{\text{mix}}), \sqrt{B_{\text{rbe}}})^2,$$

and the result follows. $\square$