

Tighter and Friendlier Integer Bounds for Quaternion Algorithms

Application to SQIsign

Maciej Czaprynko^{id}, Anisha Mukherjee^{id} and Sujoy Sinha Roy^{id}

Graz University of Technology, Graz, Austria,

`{maciej.czaprynko, anisha.mukherjee, sujoy.sinharoy}@tugraz.at`

Abstract. The digital signature scheme SQIsign, currently under consideration in NIST’s call for additional post-quantum signatures, offers the smallest key and signature sizes among all candidates. Its signing procedure, however, relies on an involved arithmetic layer over quaternions, in which the objects are represented by small-dimensional integer lattices. In this layer, the intermediate integers can grow significantly larger than the final outputs. Controlling this growth is essential for fixed-precision and, ultimately, constant-time implementations. In this work, we revisit the integer-size analysis of this arithmetic layer. We first observe that all lattices arising in SQIsign satisfy a structural containment property that allows their canonical form to be computed with a modulus of half the bit length used in previous analyses. Exploiting the same property, we give a matrix inversion algorithm using only exact divisions whose intermediate integers remain bounded by twice the modulus. This algorithm avoids the cubic blow-up of standard inversion methods. We show that the triangular-form convention used in the SQIsign implementation reduces the size of intermediate values by a factor of $O(p)$. Second, we show that these lattices can be represented in dimension two over the Gaussian integers instead of dimension four over the integers. This allows us to replace the LLL algorithm by the simpler Lagrange reduction that has the number of iterations bounded by a multiple of the bit-size of the longest input vector. Furthermore, this reduction provably outputs the shortest basis without requiring floating-point arithmetic. Combining these results, we derive integer-size bounds for SQIsign key generation and signing, improving on previous bounds for key generation and for all signing subroutines outside of the response quaternion sampling. We provide an implementation of our algorithms in the GMP-based C reference implementation of SQIsign.

Keywords: SQIsign · isogeny-based cryptography · quaternion algebras · lattice reduction

1 Introduction

Digital signatures are a prerequisite in modern cryptographic infrastructure, providing the means for secure authentication, integrity, and non-repudiation. With the transition to post-quantum cryptography (PQC) gaining traction, the National Institute of Standards and Technology (NIST) has undertaken significant efforts to standardize secure and efficient post-quantum signature schemes. While lattice-based signatures have emerged as the most mature family of PQC signatures and have already led to standardized schemes, diversification of underlying hardness assumptions remains an important objective [ABC⁺26]. In this context, isogeny-based signatures relying on the hardness of finding isogenies between supersingular elliptic curves (or equivalently, computing endomorphism rings) are a promising alternative offering compact key and signature sizes. Among these signatures,

SQIsign [AAA⁺25] is the only isogeny-based candidate submitted to the ongoing NIST call for additional digital signatures. SQIsign has the smallest key and signature sizes among all PQC signatures in the NIST call, but a comparatively involved signing procedure. The scheme relies on a close interaction between isogeny computations and quaternion ideal arithmetic enabled by the Deuring correspondence. Informally, the Deuring correspondence relates supersingular elliptic curves over $\mathbb{F}_{p^2}$ to maximal orders in a quaternion algebra $\mathcal{B}_{p,\infty}$ ramified at p and infinity. Under this correspondence, isogenies between supersingular curves can be represented algebraically by ideals between the corresponding quaternion orders. Thus, operations concerning construction or manipulation of isogenies can often be translated into those on quaternion ideals. Conversely, suitable quaternion ideals can be converted back into isogenies. At a high level, signing in these schemes involves the secret in the quaternion domain, requiring secret-dependent manipulation of ideals in quaternion orders, computation of equivalent ideals with suitable norm properties and translation between ideal-theoretic and isogeny-theoretic objects. These operations make these schemes conceptually elegant, but they also make their implementation substantially more involved than that of many other post-quantum signature schemes.

Motivation and related work. Major implementation challenges arise in the context of quaternion ideals being represented through lattices. An ideal in a quaternion order can be represented as a rank-four $\mathbb{Z}$ -lattice which usually stores a rational basis together with a common denominator. Operations on ideals therefore translate into lattice operations such as lattice basis reduction, multiplication, intersection, and conversion to canonical representations like the Hermite Normal Form (HNF). Although these lattices have small rank, their entries can become large during intermediate computations. As a result, practical implementations need careful control over the size of the integers appearing throughout the quaternion layer.

This issue is particularly important for fixed-precision and constant-time implementations. It is not sufficient to know that the final signature or final ideal representation has ‘small’ sizes, but one must also bound all intermediate values that arise during the computation. Recent works on constant-time integer arithmetic [KMJK25], 4D lattice reduction [HKK⁺25], and constant-time quaternion arithmetic [BHJ⁺25] in the context of SQIsign have also reported encountering the problem of intermediate integer size blowups, especially during HNF or greatest common divisor (GCD) computations. The work by [KLKL26] was the first to provide an upper bound on integer sizes with respect to SQIsign’s quaternion arithmetic. A follow-up work [KLY26] has demonstrated a significant reduction in these bounds at the cost of replacing HNF computations with 4-dimensional lattice reduction with up to 16 basis elements.

We note two concurrent works [ESW⁺W, LS], both of which remain unpublished at the time of writing, that are mentioned in the most recent SQIsign specifications [AAA⁺26]. As described in the specifications, [ESW⁺W] introduces the same dimension reduction as presented in this work but performs the reduction with a different algorithm. On the other hand, [LS] replaces the HNF computations with just multiplication and modular reduction, which achieve the same bounds as in our case. This comes at the cost of sampling ideals of a specific form.

1.1 Our contribution

We revisit the integer-size analysis in the context of quaternion ideal arithmetic and make the following contributions:

- **Tighter integer-size bounds in Section 3.1:** We provide improved upper bounds on the sizes of integers arising in the core quaternion subroutines of SQIsign. Our key observation is that all ideals I in SQIsign satisfy $m \cdot \mathbb{Z}^4 \subset \Lambda_I$, where m is the

norm of I or of the ideals from which it is constructed. This allows the modular HNF to be computed with a modulus of half the bit length used in [KLKL26], and enables a matrix inversion using only exact divisions whose intermediate integers stay bounded by $2m$, avoiding the cubic blowup of adjugate-based inversion. We also justify the (previously unexplained) choice of upper-triangular HNF in the SQIsign implementation by proving that, for prime-norm ideals, it reduces the Gram matrix entries by a factor of $O(p)$ compared to the lower-triangular form.

- **Dimension-reduced lattice reduction for quaternion ideals in Section 3.2:** We further exploit the structure of the starting order $\mathcal{O}_0$. Since $\mathcal{O}_0$ contains the element i , it and its one-sided ideals can be represented as two-dimensional lattices over the Gaussian integers $\mathbb{Z}[i]$ rather than four-dimensional lattices over $\mathbb{Z}$. This allows us to replace some of the LLL reductions by Lagrange reduction over $\mathbb{Z}[i]$, which provably outputs a Minkowski-reduced basis and, by adapting the analysis of [NS04], terminates in a number of iterations proportional to the bit-size of the longest input vector.
- **Application to SQIsign and C-implementation in Section 4:** We incorporate our algorithms in the GMP-based C reference implementation of SQIsign, available at: https://github.com/mczaprynko/smaller_integers_sqisign

2 Background

2.1 Notation

Matrices are denoted by bold uppercase letters, such as $\mathbf{A}$; vectors by bold lowercase letters, such as $\mathbf{a}$; and scalars by lowercase letters, such as a . For a matrix $\mathbf{A} \in R^{n \times m}$ in a ring R , the entry in row r and column c is written as $\mathbf{A}[r, c]$. We denote the r -th row by $\mathbf{A}[r, :]$, or simply $\mathbf{A}[r]$, and the c -th column by $\mathbf{A}[:, c]$. We use the notation $\mathbf{A}[r_1 : r_2]$ to denote the matrix $\mathbf{A}$ restricted to the rows between the r_1 -th row inclusive and the r_2 -th row exclusive. The same notation applies to columns.

2.2 Quaternion algebra and Deuring correspondence

We recall the quaternion background used in isogeny-based signatures. The quaternion algebra that appears in the theory of supersingular isogenies is the algebra $\mathcal{B}_{p,\infty}$ ramified exactly at a prime p and at infinity. The canonical basis of this algebra is $\{1, i, j, ij = k\}$ and it satisfies $i^2 = -1, j^2 = -p, ij = -ji$. Any quaternion element can be written in this basis as $\alpha = a + bi + cj + dk \in \mathcal{B}_{p,\infty}$ with $a, b, c, d \in \mathbb{Q}$. The conjugate is $\bar{\alpha} = a - bi - cj - dk \in \mathcal{B}_{p,\infty}$.

Quaternion norms. An equivalent to the usual Euclidean norm $\|\cdot\|$ and scalar product $\langle \cdot, \cdot \rangle$ can be defined for the quaternion algebra. For $\alpha, \beta \in \mathcal{B}_{p,\infty}$, the equivalent of the scalar product is the trace $\text{Tr}(\alpha, \beta) = \alpha\bar{\beta} + \beta\bar{\alpha}$ and the equivalent to the norm $\|\cdot\|^2$ is the reduced norm $\text{nrd}(\alpha) = \alpha\bar{\alpha}$.

Orders and ideals. A maximal order $\mathcal{O} \subset \mathcal{B}_{p,\infty}$ is a subring containing 1 that is also a full-rank $\mathbb{Z}$ -lattice in $\mathcal{B}_{p,\infty}$ and is not properly contained in any larger order. A commonly used maximal order is $\mathcal{O}_0$, generated by $(1, i, (1+k)/2, (i+j)/2)$. Ideals of quaternion orders play the role of algebraic counterparts of supersingular isogenies. A left (right) integral $\mathcal{O}$ -ideal I is a finitely generated $\mathbb{Z}$ -sublattice of $\mathcal{O}$ that is closed under left (right) multiplication by $\mathcal{O}$. Associated to I are its left and right orders, $\mathcal{O}_L(I) = \{\alpha \in \mathcal{B}_{p,\infty} : \alpha I \subset I\}$, $\mathcal{O}_R(I) = \{\alpha \in \mathcal{B}_{p,\infty} : I\alpha \subset I\}$. As such, an ideal I is said

to be a left $\mathcal{O}$ -ideal if $\mathcal{O}_L(I) = \mathcal{O}$, similarly for right $\mathcal{O}$ -ideals. If the order being on the left or the right is not important, we simply say an ideal is an $\mathcal{O}$ -ideal.

The norm of an ideal $\text{nrd}(I)$ is defined as the GCD of the reduced norms of its elements, $\text{nrd}(I) = \gcd(\{\text{nrd}(\alpha) : \alpha \in I\})$. If the norm of I satisfies $\text{nrd}(I) \in \mathbb{Z}$, the ideal is said to be integral and we assume hereafter that every ideal is integral unless stated otherwise. Such an ideal I can be generated by a suitable primitive element in I along with the norm of the ideal, that is, $I = \mathcal{O}\alpha + \mathcal{O}\text{nrd}(I)$, $\alpha \in I$. Two quaternion ideals I and J are said to be equivalent if there exists $\alpha \in \mathcal{B}_{p,\infty}$ such that $J = I\alpha$. We define the map χ_I from I to J by $\chi_I(\alpha) = I\bar{\alpha}/\text{nrd}(I)$. If $\alpha \in I$, then $\chi_I(\alpha)$ corresponds to a primitive equivalent ideal.

Deuring correspondence. The Deuring correspondence [Deu41] identifies the endomorphism ring of a supersingular elliptic curve in characteristic p with a maximal order in $\mathcal{B}_{p,\infty}$, and identifies an isogeny $\varphi: E_0 \rightarrow E_1$ with an ideal that is a left $\text{End}(E_0)$ -ideal and a right $\text{End}(E_1)$ -ideal, so that isogeny operations translate into ideal operations. We refer to [AAA⁺25] for further details.

Quaternion ideal arithmetic. In the next section, we define the addition, intersection, and multiplication of ideals in terms of lattice operations. Furthermore, we also define basis reduction and the HNF. These last two operations do not change the underlying ideal. Instead, they provide useful properties for more complex operations.

2.3 Lattices and lattice reduction algorithms

A lattice Λ of rank r is a discrete subgroup of $\mathbb{R}^n$. In other words, $\Lambda = \{\sum_{i=1}^r c_i \mathbf{b}_i : c_i \in \mathbb{Z}, i \in \{1, 2, \dots, r\}\}$. Thus a lattice consists of all $\mathbb{Z}$ -linear combinations of r linearly independent vectors $\mathbf{b}_i \in \mathbb{R}^n$. The vectors $\mathbf{b}_i$ comprise a ‘basis’ of the lattice. A basis can also be represented by a matrix $\mathbf{B}$ of column vectors such that $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_r)$. The volume, or covolume, of Λ is invariant under basis change. If $\mathbf{G}$ is the Gram matrix, then $\text{Vol}(\Lambda) = \sqrt{\det(\mathbf{G})}$. For bases that form a full-rank matrix $\mathbf{B}$, we have that $\text{Vol}(\Lambda) = |\det(\mathbf{B})|$. While the volume is fixed, the lengths and angles of the basis vectors may vary significantly between different bases. Often, one is required to transform a given lattice basis into a ‘nice’ basis for the lattice, that is, to a basis with a lesser defect. In dimension 4, a Minkowski-reduced basis describes a good basis in the sense that it contains the shortest vectors of the lattice [vdW]. Lattice basis reduction refers to the techniques used for such a transformation to a basis consisting of short and close to orthogonal vectors. Reduction of two-dimensional lattice bases was given by Lagrange and Gauss. In Section 3.2, we provide pseudo-code for the two-dimensional lattice reduction which we denote as ‘Lagrange Reduction’ hereafter. For higher-dimension lattices, the notion of LLL-reduced basis [LLL82] is often used and is detailed hereafter.

Gram-Schmidt orthogonalization (GSO). Given a set of basis vectors $\mathbf{b}_i \in \mathbb{R}^n, \forall i \in \{1, 2, \dots, r\}$, the GSO process computes an orthogonal basis $\mathbf{b}_i^\perp \in \mathbb{R}^n, \forall i \in \{1, 2, \dots, r\}$ that spans the same subgroup as the input vectors. The vectors are computed iteratively as follows:

$$\mathbf{b}_i^\perp = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \mathbf{b}_j^\perp, \text{ with } \mu_{i,j} = \frac{\langle \mathbf{b}_i, \mathbf{b}_j^\perp \rangle}{\|\mathbf{b}_j^\perp\|^2}. \quad (1)$$

Notice that the lattice generated by the resulting basis is generally different from the one generated by the input basis since $\mu_{i,j}$ can take values in $\mathbb{R}$.

LLL reduction. Introduced in [LLL82], the LLL algorithm is a polynomial-time algorithm that computes a (δ, η) -LLL reduced basis, for $\delta \in (1/4, 1)$ and $\eta \in [1/2, \sqrt{\delta})$. Concretely, a (δ, η) -LLL reduced basis $(\mathbf{b}_1, \dots, \mathbf{b}_r)$ satisfies [NS09]:

- (Lovász condition): $\delta \|\mathbf{b}_{\kappa-1}^\perp\|^2 \leq \|\mathbf{b}_\kappa^\perp + \mu_{\kappa,\kappa-1} \mathbf{b}_{\kappa-1}^\perp\|^2$, for all $2 \leq \kappa \leq r$,
- (Size-reduction): $|\mu_{i,j}| \leq \eta$, for all $1 \leq j < i \leq r$.

Furthermore, the product of the norms of the output basis is upper-bounded as:

$$\|\mathbf{b}_1\| \|\mathbf{b}_2\| \dots \|\mathbf{b}_r\| \leq (1/(\delta - \eta^2))^{\frac{r(r-1)}{4}} \text{Vol}(\Lambda_B) \quad (2)$$

where Λ_B is the lattice defined by the basis $(\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_r)$.

To compute the (δ, η) -LLL reduced basis, SQIsign uses the L2 algorithm [NS09] which is a floating-point implementation of the LLL algorithm. It also achieves a (δ, η) -LLL reduced basis, with the limitation that $\eta > 1/2$. The pseudo-code and an integer bound analysis are provided in Algorithm 12.

2.3.1 Quaternion lattices and lattice operations

A quaternion lattice is generated by four $\mathbb{Q}$ -linearly independent quaternion elements, $\Lambda = \alpha_1\mathbb{Z} + \alpha_2\mathbb{Z} + \alpha_3\mathbb{Z} + \alpha_4\mathbb{Z}$. After fixing the basis $(1, i, j, k)$, the basis elements $\alpha_1, \dots, \alpha_4$ are stored as the columns of a matrix $\mathbf{B}$ so that $(\alpha_1 \ \alpha_2 \ \alpha_3 \ \alpha_4) = (1 \ i \ j \ k) \cdot \mathbf{B}$. We now define some of the main lattice operations required for describing our contributions.

Addition. The sum $\Lambda_1 + \Lambda_2$ is generated by the union of the two sets of basis vectors. Matrix-wise, one concatenates the basis matrices as $\mathbf{B}_1 \parallel \mathbf{B}_2$ and then computes a basis of the resulting lattice (for example, by using HNF in SQIsign).

Dual and intersection. The dual lattice is defined by $\Lambda^* = \{f \in \mathcal{B}_{p,\infty}^* : f(x) \in \mathbb{Z} \text{ for all } x \in \Lambda\}$, where $\mathcal{B}_{p,\infty}^*$ is the space of linear functions $\mathcal{B}_{p,\infty} \rightarrow \mathbb{Q}$. If Λ is represented by $(1 \ i \ j \ k) \cdot \mathbf{B}$, then the dual lattice is represented in the dual coordinate basis $(1^* \ i^* \ j^* \ k^*)$ multiplied by $(\mathbf{B}^{-1})^\top$. The intersection of two lattices can then be computed through duals: $\Lambda_1 \cap \Lambda_2 = (\Lambda_1^* + \Lambda_2^*)^*$. Thus, computing the intersection of two lattices reduces to computing the inverse of two lattice bases, a lattice sum, and a second inverse.

Multiplication. If Λ_1 ($\mathbf{B}_1$ in matrix form) and Λ_2 ($\mathbf{B}_2$ in matrix form) are quaternion lattices, their product is the lattice generated by all products $\mathbf{x}\mathbf{y}$, with $\mathbf{x} \in \Lambda_1$ and $\mathbf{y} \in \Lambda_2$. Concretely, if $\alpha_1, \dots, \alpha_4$ is a basis of the lattice corresponding to $\mathbf{B}_2$, and $\mathbf{A}_1, \dots, \mathbf{A}_4$ are the matrices describing right multiplication by these basis elements, then $\Lambda_1\Lambda_2 = \mathbf{A}_1\mathbf{B}_1 + \dots + \mathbf{A}_4\mathbf{B}_1$. In practice, the multiplications by the matrices $\mathbf{A}_r, \forall r \in \{1, \dots, 4\}$ are performed with the quaternion representation. The product is therefore computed using quaternion multiplication followed by a lattice sum.

Gram matrix. For the standard basis $(1, i, j, k)$, the corresponding Gram matrix is $\mathbf{G} = \text{diag}(1, 1, p, p)$. If $\mathbf{M}$ is a change-of-basis matrix, then the Gram matrix of any other transformed basis is $\mathbf{G}' = \mathbf{M}^\top \mathbf{G} \mathbf{M}$. This is the form used by lattice reduction algorithms in the quaternion setting. Equivalently, the Gram matrix of a quaternion lattice Λ can be computed using the basis matrix $\mathbf{B}$ as $\mathbf{G} = \mathbf{B}^\top \text{diag}(1, 1, p, p) \mathbf{B}$.

Hermite Normal Form (HNF). The HNF of a matrix can be seen as an analogue of row-echelon form over $\mathbb{Z}$. SQIsign [AAA⁺25] uses the column-style convention: if $\mathbf{A} \in \mathbb{Z}^{d \times c}$ has rank d , then its HNF is a matrix $\mathbf{H} \in \mathbb{Z}^{d \times c}$ such that $\mathbf{H} = \mathbf{A} \cdot \mathbf{U}$ for some unimodular matrix $\mathbf{U}$ with integer coefficients. Here we differentiate between two conventions on the HNF: an upper- and a lower-HNF, according to whether the matrix is upper- or lower-triangular. As defined, the upper-HNF corresponds to SQIsign's definition of the HNF. For the upper-HNF, the nonzero part of $\mathbf{H}$ is triangular, its pivots are positive, zero

columns, if present, are placed to the left, the pivot positions move strictly to the right as one moves down the rows and the entries to the right of a pivot are non-negative and strictly smaller than the pivot. A lower-HNF can be defined similarly.

3 Improved quaternion lattice algorithms

In this section, we discuss our improved quaternion algorithms that are used to prove the new integer bounds on the quaternion subroutines of SQIsign later in Section 4. The algorithms are discussed in Section 3.1 where we derive the upper bounds on the integer sizes. One important property that we use is related to Minkowski-reduced bases, which are not guaranteed to be obtained using LLL (or BKZ)-based reductions. Hence, in Section 3.2, we show that a Minkowski-reduced basis can be computed efficiently using a Lagrange reduction algorithm by reducing the dimension of lattices corresponding to $\mathcal{O}_0$ -ideals from 4 to 2 dimensions. However, this comes at the cost of working over Gaussian integers.

3.1 Integer size reduction and bound improvements

In this section, we prove the integer bounds of three subroutines used in SQIsign to perform ideal, and thus lattice, operations. First, in Section 3.1.1 we reduce the upper bound of the HNF algorithm, which is mainly involved in the creation of ideals, ideal addition and ideal multiplication. Next, in Section 3.1.2 dual ideals are studied. In particular, we provide an algorithm that computes the inverse of an ideal I . From that, we are able to analyze the sizes of the integers in the inverse lattice and the corresponding Gram matrix of dual ideals. The main observation used to obtain low bounds for the computation is that the order $\mathcal{O}_0$, defined in Section 2, contains the quaternion elements $(1, i, j, k)$. Moreover, every left $\mathcal{O}_0$ -ideal I can be written as $I = \mathcal{O}_0\alpha + \mathcal{O}_0 \cdot \text{nrd}(I)$, for some quaternion α . A similar statement holds for right $\mathcal{O}_0$ -ideals. This implies the following property:

Property 1. Every lattice Λ obtained from an $\mathcal{O}_0$ -ideal I contains $\text{nrd}(I)\mathbb{Z}^4$.

In Section 3.1.3, we improve the upper bound of the sampling of random equivalent ideals.

3.1.1 Hermite Normal Form (HNF)

The current algorithm in SQIsign uses the modular HNF algorithm [Coh93, Section 2.4.2]. The core idea is that the HNF computation can be performed modulo the determinant (or a multiple of it) of the resulting lattice since, for an integer lattice Λ , we have that $\det(\Lambda) \cdot \mathbb{Z}^4 \subset \Lambda$. This implies $\Lambda + \det(\Lambda) \cdot \text{Id}_4 = \Lambda$ and we can thus concatenate the multiple of the identity matrix to our basis vectors that need to be reduced. This is essentially equivalent to working modulo $\det(\Lambda)$ since at each point we can subtract a multiple of the newly added columns to our intermediate results except for the current pivot. This approach requires checking at each iteration that the pivot has not been reduced. The algorithm to compute the upper-HNF is presented in Algorithm 1. In the algorithm, `ExtGCD` is the extended GCD that takes as input two integers $a, b \in \mathbb{Z}$ and returns their GCD $g \in \mathbb{Z}$ and 4 values $u, v, s, t \in \mathbb{Z}$ such that $g = u \cdot a + v \cdot b$, $0 = s \cdot a + t \cdot b$, and the absolute values $|u|, |v|, |s|, |t|$ are the smallest possible. This corresponds to the Bézout coefficients and a pair of the form $(-b/g, a/g) = (s, t)$. For these values we have that $|u|, |s| \leq |b|$ and $|v|, |t| \leq |a|$.

Algorithm 1 HNFModm($\mathbf{L}, m$), based on [Coh93, Section 2.4.2]

Input: The matrix $\mathbf{L} \in \mathbb{Z}^{4 \times n}$, $n \geq 4$ contains the column-wise basis elements of a lattice Λ and m is an element such that $m \neq 0$ and $m\mathbb{Z}^4 \subseteq \Lambda$.

Output: Result is a 4×4 matrix in upper-HNF.

```

1:  $\mathbf{W} \leftarrow \mathbf{L} \pmod m$ 
2: for  $r = 4$  downto 1 do
3:    $piv \leftarrow n + r - 4$ 
4:   for  $c = 1$  to  $piv - 1$  do
5:      $(gcd, u, v, s, t) \leftarrow \text{ExtGCD}(\mathbf{W}[r, piv], \mathbf{W}[r, c])$  // Assumes that  $(1, 1, 0, 0, 1) = \text{ExtGCD}(0, 0)$ 
6:      $\mathbf{w} \leftarrow u \cdot \mathbf{W}[:, piv] + v \cdot \mathbf{W}[:, c]$ 
7:      $\mathbf{W}[:, c] \leftarrow s \cdot \mathbf{W}[:, piv] + t \cdot \mathbf{W}[:, c]$ 
8:      $\mathbf{W}[:, piv] \leftarrow \mathbf{w}$ 
9:      $\mathbf{W}[r, piv] \leftarrow \mathbf{W}[r, piv] \pmod m$ 
10:     $\mathbf{W}[r, c] \leftarrow \mathbf{W}[r, c] \pmod m$ 
11:     $(gcd, u, v, s, t) \leftarrow \text{ExtGCD}(\mathbf{W}[r, piv], m)$ 
12:     $\mathbf{W}[:, piv] \leftarrow u \cdot \mathbf{W}[:, piv] \pmod m$ 
13:    if  $\mathbf{W}[r, piv] = 0$  then
14:       $\mathbf{W}[r, piv] = m$ 
15:    for  $c = piv + 1$  to  $n$  do
16:       $q \leftarrow \lfloor \mathbf{W}[r, c] / \mathbf{W}[r, piv] \rfloor$ 
17:       $\mathbf{W}[(r + 1), c] \leftarrow (\mathbf{W}[(r + 1), c] - q \cdot \mathbf{W}[(r + 1), piv]) \pmod m$ 
18: return  $\mathbf{W}[:, n - 3 : n + 1]$ 

```

In the case of a quaternion ideal I , the corresponding lattice has determinant $\text{nrd}(I)^2/4$. In our approach, we use the fact that the lattice operations in SQIsign always involve $\mathcal{O}_0$ -left or right ideals whenever the HNF is used. This fact is useful since in that case, $\text{nrd}(I) \cdot \mathbb{Z}^4 \subset I$. This is a consequence of Property 1: all $\mathcal{O}_0$ -ideals contain $\mathcal{O}_0 \cdot \text{nrd}(I)$ and thus $\text{nrd}(I) \cdot \mathbb{Z}^4$ since $\mathbb{Z} + \mathbb{Z} \cdot i + \mathbb{Z} \cdot j + \mathbb{Z} \cdot k \subset \mathcal{O}_0$. To work over integer values, we need to set each element to the same denominator. In the $\mathcal{O}_0$ case, the denominator is 2, meaning that we can perform the HNF modulo $2 \cdot \text{nrd}(I)$. As such, we have the following lemma on the bounds for the integers present in HNF:

Lemma 1. *Let $\mathbf{L} \in \mathbb{Z}^{4 \times n}$, $n \geq 4$ be a matrix that contains the column-wise basis elements of a lattice Λ and let m be an integer such that $m \neq 0$ and $m\mathbb{Z}^4 \subseteq \Lambda$. The integer sizes of the computations in Algorithm 1 are upper-bounded by m^2 and the resulting HNF matrix has positive coefficients of size at most m .*

Proof. In Algorithm 1 it is clear that the integers cannot go beyond the modulus m squared. The modulus itself follows from the definitions of the operations in Section 2 and Property 1. $\square$

From Lemma 1, the operations defined in Section 2 and Property 1, we obtain the following Corollary 1. Note that for equivalent ideals, $\text{nrd}(I)$ divides the basis elements of the equivalent ideal. As such, it is assumed that the fractions are simplified before computing the HNF.

Corollary 1. *Let I_i for $i \in \{1, 2\}$ be some input $\mathcal{O}_0$ -ideals and I_o be an output $\mathcal{O}_0$ -ideal. Then, the HNF computations during the ideal creation and equivalent ideal computation are upper-bounded by $(2 \cdot \text{nrd}(I_o))^2$, the ideal addition by $(2 \cdot \max(\text{nrd}(I_1), \text{nrd}(I_2)))^2$, and the ideal multiplication by $(4 \cdot \text{nrd}(I_1) \text{nrd}(I_2))^2$.*

Lower norms of the Gram matrix. Notice that $\text{nrd}(x) = x_1^2 + x_2^2 + p \cdot (x_3^2 + x_4^2)$ is unbalanced towards the last two elements. We would prefer to keep them as small as possible. As such, upper-triangular HNF is better than the lower-triangular one. The reverse is true for dual lattices since the basis is $(1, i, j/p, k/p)$. Heuristically, this reduces the size of norms by a factor of $O(p)$. Note that while the reason for the choice is not explicitly mentioned, the upper-triangular form is implemented in SQIsign.

For ideals of large-enough prime norms, the reduction is provable. The author of [Ler26, §3.2]¹ shows that ideals with prime norms have the following HNF basis in upper and lower triangular form respectively:

$$2 \cdot L_U = \begin{pmatrix} 2N & 0 & x & y \\ 0 & 2N & 2N - y & x \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad 2 \cdot L_L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ x' & 2N - y' & 2N & 0 \\ y' & x' & 0 & 2N \end{pmatrix}, \quad (3)$$

for some $0 \leq x, y, x', y' < 2N$, where N is a prime in $\mathbb{N}$. It follows that the reduced norm of a vector in L_U is bounded by $8N^2 + p$ while for L_L it is $2 + 8p \cdot N^2 \approx 8p \cdot N^2$. Assuming that $N \approx \sqrt{p}$, we have that the ratio between the two representations is approximately $\frac{8}{9}p$ for prime-norm ideals. For non-prime norms, the structure of the matrix is more complicated (see for example [Cle25] for ideals that have a prime power norm) which we do not analyze here in more detail. Hence we rely on the general result from Lemma 1. From the above, we have the following lemma:

Lemma 2. *$\mathcal{O}_0$ -ideals I of prime norm have their HNF as in Eq. (3) and the integer sizes of the Gram matrix of the basis of I in upper-HNF are upper-bounded by $8N^2 + p$. For the composite norms, the upper bound is $16p \cdot \text{nrd}(I)^2$.*

3.1.2 Dual ideals

Integer bounds for lattice inversion. The inversion of a lattice basis $\mathbf{B}$ is used in SQIsign to compute the intersection of ideals. A common approach is to compute it as

$$\mathbf{B}^{-1} = \frac{\text{adj}(\mathbf{B})}{\det(\mathbf{B})}. \quad (4)$$

While the determinant can be precomputed from the properties of quaternion ideals, the size of integers during the computation of the adjugate can grow to the size of the cube of the inputs. For ideals and the corresponding lattices that satisfy Property 1, we propose the following lemma:

Lemma 3. *Let $\mathbf{B} \in \mathbb{Z}^{4 \times 4}$ be in upper-HNF and let $m \in \mathbb{Z}$ such that $m\mathbb{Z}^4 \subset \Lambda_{\mathbf{B}}$, where $\Lambda_{\mathbf{B}}$ is the lattice spanned by the column vectors of $\mathbf{B}$. Then the denominator of the inverse is m and the integer part of the inverted matrix has entries of at most $2m$. Furthermore, the integer sizes in Algorithm 2 that computes the inverse of $\mathbf{B}$ are bounded by $2m$.*

Proof. The first statement follows from the fact that since $m\mathbb{Z}^4 \subset \Lambda_{\mathbf{B}}$, there exists a matrix $\mathbf{M} \in \mathbb{Z}^{4 \times 4}$ such that:

$$\mathbf{B}\mathbf{M} = m \cdot \mathbf{Id}_4. \quad (5)$$

It follows that $\mathbf{M}/m = \mathbf{B}^{-1}$. Next, we prove the statements on the integer sizes of $\mathbf{B}^{-1}$ and Algorithm 2. For this, we consider the computation of the inverse from the left side of $\mathbf{B}$ which yields the same inverse. Furthermore, since the matrix is in upper-HNF, we have that the diagonal elements of $\mathbf{B}$ are $\leq m$ and that the upper elements are smaller than m . Given these properties, we analyze the first iteration of the outer loop in Algorithm 2 which computes the first row of the inverse. The remaining iterations can be deduced from it.

During the computation, the first element on the first row of the inverse, $\mathbf{W}[1, 1]$, depends only on $\mathbf{B}[1, 1]$ as the rest of the column is composed of zeros. Under our assumptions, this element must divide m otherwise $m\mathbb{Z}^4 \not\subset \Lambda_{\mathbf{B}}$. The result of the division

¹Note that the author provides more efficient HNF computation and intersection algorithms. They are not generic, however: the HNF requires the ideal to have prime norm. As for the intersection, it requires the norms to be coprime and their factorization to be known.

is thus $\mathbf{W}[1, 1]$. For the next element, we can apply the same reasoning on the second column, this time it depends on the first two elements of the column since the corresponding entry of $\mathbf{WB}$, obtained as a sum of products, must be zero. There, the only unknown is the second element of the inverse row since we computed the first one at this point. It follows that the element can be computed as $-\mathbf{B}[1, 2] \cdot \mathbf{W}[1, 1] / \mathbf{B}[2, 2]$. With the same logic, the four elements can be computed as

$$\mathbf{W}[1, 1] = m / \mathbf{B}[1, 1], \quad \mathbf{W}[1, c] = -\mathbf{v}^{(c-1)}[c] / \mathbf{B}[c, c], \quad (6)$$

where

$$\mathbf{v}^{(1)} = \mathbf{W}[1, 1] \cdot \mathbf{B}[1, 1], \quad \mathbf{v}^{(c)} = \mathbf{v}^{(c-1)} + \mathbf{W}[1, c] \cdot \mathbf{B}[c, c], \quad (7)$$

and $c \in \{1, \dots, 4\}$. The computational bounds can be derived from Eq. (6) and Eq. (7) and from the fact that the basis is in HNF. Concretely, we have that each element in $\mathbf{v}^{(1)}$ is positive and bounded by m since $\mathbf{W}[1, 1] \cdot \mathbf{B}[1, 1] = m$ and $0 \leq \mathbf{B}[1, c] < \mathbf{B}[1, 1], \forall c \in \{2, 3, 4\}$. For the next iteration, we have that $0 \geq \mathbf{W}[1, 2] \cdot \mathbf{B}[2, 2] = -\mathbf{v}^{(1)}[2] > -m$ and $0 \leq \mathbf{B}[2, c] < \mathbf{B}[2, 2], \forall c \in \{3, 4\}$, hence the absolute value of the elements of the vector $\mathbf{v}^{(2)}$ is bounded by m . By the same logic, the absolute value of the elements of $\mathbf{v}^{(3)}$ is bounded by $2m$.

The remaining rows of the inverted matrix are obtained using the same method, but with the observation that the first r coefficients will be zero. As such, during the computation, the integers do not go beyond $2m$. Moreover, since $\mathbf{B}_{inv}$ is triangular and the diagonal elements divide m , the off-diagonal elements can be taken modulo m if needed. The resulting matrix thus can be made to have elements between 0 and m . $\square$

Algorithm 2 MatrixInversion($\mathbf{B}, m$)

Input: A matrix $\mathbf{B} \in \mathbb{Z}^{4 \times 4}$ in HNF and an integer m such that $m\mathbb{Z}^4$ is in the lattice spanned by $\mathbf{B}$.

Output: The inverse of the matrix $\mathbf{B}_{inv} = \mathbf{B}^{-1}$

```

1:  $\mathbf{T} \leftarrow \mathbf{B}$ 
2:  $\mathbf{W} \leftarrow \mathbf{0}^{4 \times 4}$ 
3: for  $r \leftarrow 1$  to 4 do
4:    $\mathbf{W}[r, r] \leftarrow m / \mathbf{T}[r, r]$  // The division is exact
5:    $\mathbf{v}^{(1)} \leftarrow \mathbf{W}[r, r] \cdot \mathbf{T}[r]$ 
6:   for  $c \leftarrow r + 1$  to 4 do
7:      $\mathbf{W}[r, c] \leftarrow -\mathbf{v}^{(c-r)}[c] / \mathbf{T}[c, c]$  // The division is exact
8:      $\mathbf{v}^{(c-r+1)} \leftarrow \mathbf{v}^{(c-r)} + \mathbf{W}[r, c] \cdot \mathbf{T}[c]$ 
9:  $\mathbf{B}_{inv} \leftarrow \mathbf{W} / m$ 
10: return  $\mathbf{B}_{inv}$ 
```

Integer bounds for ideal intersection. We propose the following lemma:

Lemma 4. *Let I_1 and I_2 be two $\mathcal{O}_0$ -ideals with their basis vectors in upper-HNF. The ideal intersection requires integers of size at most $(2 \cdot \text{nrd}(I_1) \cdot \text{nrd}(I_2))^2$.*

Proof. Let $\mathbf{B}_1$ and $\mathbf{B}_2$ be the lattice bases containing the basis elements of I_1 and I_2 respectively. We recall from Section 2.3 that the basis of the intersection of the ideals, $\mathbf{B}_{I_1 \cap I_2}$, is obtained from the lattices computed as:

$$\Lambda_{I_1 \cap I_2} = (\Lambda_1^* + \Lambda_2^*)^*, \quad (8)$$

where the lattices correspond to the previously defined bases. Therefore, the computation is performed as two inversions followed by a lattice addition and finally another inversion. From Lemma 3, the denominator of $\mathbf{B}_i^{-1}$ is $\text{nrd}(I_i)$ and we have that $2 \cdot \mathbb{Z}^4 \subset \Lambda_i^*$, for all $i \in \{1, 2\}$. It follows from Lemma 1 that the addition can be performed with integer sizes at most $2 \cdot \text{nrd}(I_1) \cdot \text{nrd}(I_2)$ after setting the basis elements to the common denominator. Furthermore, for the same reason as previously, we have from Lemma 3 that the inversion step can be performed with integer sizes bounded by $4 \cdot \text{nrd}(I_1) \cdot \text{nrd}(I_2)$. As such, the integer sizes during the overall computation are upper-bounded by $(2 \cdot \text{nrd}(I_1) \cdot \text{nrd}(I_2))^2$. $\square$

Bound on the norm of dual vectors. We propose the following lemma:

Lemma 5. *Let I be an $\mathcal{O}_0$ -ideal. The reduced norms of the quaternions $\alpha \in I^\star$ admit a representation with a denominator equal to $p \cdot \text{nrd}(I)$ as does the Gram matrix of the dual ideal. Let $\beta = (p \cdot \text{nrd}(I))\alpha$ be the integer part of α . We have equivalently that the reduced norm of β is divisible by $p \cdot \text{nrd}(I)$.*

Proof. Let $\mathbf{G}_I$ be the Gram matrix of the lattice containing the basis elements of an $\mathcal{O}_0$ -ideal I and let $\mathbf{G}_I^\star$ be the Gram matrix of the dual ideal $I^\star$. Since the Gram matrix of an ideal is divisible by the norm of the ideal and $\det(\mathbf{G}_I) = p^2 \text{nrd}(I)^4/16$, we have that:

$$\mathbf{G}_I^\star = \mathbf{G}_I^{-1} = \left(\frac{\text{nrd}(I)}{2} \mathbf{G}'_I \right)^{-1} = \frac{2}{\text{nrd}(I)} \frac{\text{adj}(\mathbf{G}'_I)}{\det(\mathbf{G}'_I)} = \frac{\text{adj}(\mathbf{G}'_I)}{2p^2 \text{nrd}(I)}, \quad (9)$$

where $\mathbf{G}'_I$ is the integer part of $\mathbf{G}_I$ with the common factor $\text{nrd}(I)$ removed. Eq. (9) implies that the denominator of the norm of a dual element has at most a single multiple of $\text{nrd}(I)$. Furthermore, since the basis of the dual lattice is $(1, i, j/p, k/p)$, the reduced norm of a quaternion element $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ of a dual lattice can be computed as $p \cdot \text{nrd}(\alpha) = p \cdot (\alpha_1^2 + \alpha_2^2) + \alpha_3^2 + \alpha_4^2$. From Lemma 3, we can obtain a lattice $\mathbf{B}_I^\star / \text{nrd}(I)$ representing the basis of the dual ideal, with $\mathbf{B}_I^\star \in \mathbb{Z}^{4 \times 4}$. As such, the Gram matrix of the dual ideal can be written as:

$$\mathbf{G}_I^\star = \frac{\mathbf{B}_I^{\star\top}}{\text{nrd}(I)} \cdot \frac{\text{diag}(p, p, 1, 1)}{p} \cdot \frac{\mathbf{B}_I^\star}{\text{nrd}(I)} = \frac{\mathbf{G}_I''}{p \cdot \text{nrd}^2(I)}, \quad (10)$$

for some matrix $\mathbf{G}_I'' \in \mathbb{Z}^{4 \times 4}$. It follows from Eq. (9) and Eq. (10) that the reduced norm of the elements in the lattice $\mathbf{B}_I^\star$ is divisible by $\text{nrd}(I)$ and the denominator of the Gram matrix $\mathbf{G}_I^\star$ can be set to $p \cdot \text{nrd}(I)$. $\square$

3.1.3 Bounding the norm of equivalent ideals

Sampling a random ideal requires a relatively large norm for the distribution to be close to uniform, which makes the translation to isogenies difficult. Fortunately, an equivalent ideal of norm $O(\sqrt{p})$ can be obtained using Algorithm 3. In this section, we show that sampling an equivalent ideal from a Minkowski-reduced basis is equivalent to sampling from the smallest equivalent ideal. Combined with the heuristic on the distribution of smallest equivalent ideals from [AAA⁺25, Section 9.3.2], this yields more conservative bounds than the experimental values observed in [KLY26]. Concretely:

Lemma 6. *Let I be a left $\mathcal{O}_0$ -ideal with its basis in HNF, then the integers in Algorithm 3 reach a size of at most $2^4 p \cdot \text{nrd}(I)^2$ during the sampling of an equivalent ideal J . If $\text{nrd}(I)$ is prime, the bound becomes $\max(8 \cdot \text{nrd}(I)^2 + p, 2^{10} p \cdot \text{nrd}(I))$. Furthermore, the norm of J is bounded heuristically by $2^{48} \sqrt{p}$ while the worst-case bound is $2^{15} p$.*

Algorithm 3 RandomEquivalentPrimeIdeal(I)

Input: I , a left $\mathcal{O}_0$ -ideal with its basis in HNF.

Output: $J \sim I$ of small prime norm, or raise an exception if unsuccessful.

```

1: Initialize counter  $\leftarrow 0$ 
2: Compute a Minkowski-reduced basis  $\alpha_1, \alpha_2, \alpha_3, \alpha_4$  of  $I$ 
3: while counter  $< (2 \cdot \text{QUAT\_equiv\_bound\_coeff} + 1)^4$  do
4:   counter  $\leftarrow$  counter + 1
5:   Sample  $c_1, c_2, c_3, c_4$  uniformly at random from  $[-b, \dots, b]$ , for bound  $b = \text{QUAT\_equiv\_bound\_coeff}$ 
6:    $\beta \leftarrow \sum_{t=1}^4 c_t \alpha_t$ 
7:    $J \leftarrow \chi_I(\beta)$ 
8:   if  $\text{nrd}(J)$  is prime then
9:     return  $J$ 
10: raise Exception: ("RandomEquivalentPrimeIdeal failed")
```

Proof. We prove the lemma in two parts: first, we prove the claim on the norm of the equivalent ideal J ; then we go over the computation of Algorithm 3 step by step to derive an upper bound on the size of the integers involved.

Let $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ be the Minkowski-reduced basis of an ideal I and $\beta \in \mathcal{B}_{p,\infty}^*$ be a quaternion such that the equivalent ideal, $I_s \cdot \frac{\beta}{\text{nrd}(I_s)} = I$, is the shortest one. In that case, we have that the basis of the ideal I_s defined as:

$$\sigma_t = \alpha_t \bar{\beta} \frac{\text{nrd}(I_s)}{\text{nrd}(\beta)}, \forall t \in \{1, 2, 3, 4\}, \quad (11)$$

is Minkowski-reduced. To compute an equivalent ideal, a random linear combination of the basis elements of I is chosen as the equivalent quaternion, let it be $\gamma = \sum_{t=1}^4 c_t \alpha_t$. This is repeated until the norm of the resulting ideal I_r is prime. As such, we have that the norm of the ideal is:

$$\text{nrd}(I_r) = \frac{\text{nrd}(\gamma)}{\text{nrd}(I)} = \frac{\text{nrd}(\sum_{t=1}^4 c_t \alpha_t)}{\text{nrd}(I)} = \frac{\text{nrd}(\sum_{t=1}^4 c_t \sigma_t) \text{nrd}(\beta)}{\text{nrd}^2(I_s) \text{nrd}(I)} = \frac{\text{nrd}(\sum_{t=1}^4 c_t \sigma_t)}{\text{nrd}(I_s)}. \quad (12)$$

The last equality follows from the fact that $\text{nrd}(I) = \frac{\text{nrd}(\beta)}{\text{nrd}(I_s)}$, thus showing that sampling an ideal from I is equivalent to sampling from the shortest one I_s . To obtain the bounds on the norm, we use the triangle inequality with the fact that the basis of I_s is Minkowski-reduced to get:

$$\frac{\text{nrd}(\sigma_1)}{\text{nrd}(I_s)} \leq \frac{\text{nrd}(\sum_{t=1}^4 c_t \sigma_t)}{\text{nrd}(I_s)} \leq 16 \cdot c_{\max}^2 \frac{\text{nrd}(\sigma_4)}{\text{nrd}(I_s)}, \quad (13)$$

where $c_{\max}$ is the maximal possible coefficient. Finally, to get the concrete bounds, we restrict the analysis to $\mathcal{O}_0$ -ideals and we re-frame the bounds in terms of $\text{nrd}(\sigma_1)$ only. In that case, since the basis is Minkowski-reduced, we have the following inequality:

$$\text{nrd}(\sigma_1)^2 \text{nrd}(\sigma_4)^2 \leq \frac{p^2}{4} \text{nrd}(I_s)^4 \iff \frac{\text{nrd}(\sigma_4)}{\text{nrd}(I_s)} \leq \frac{p}{2} \frac{\text{nrd}(I_s)}{\text{nrd}(\sigma_1)}. \quad (14)$$

For the last inequality, we use the heuristic on the distribution of $\text{nrd}(I_s)/\text{nrd}(\sigma_1)$ from [AAA⁺25, Section 9.3.2]:

$$\Pr \left\{ \frac{\text{nrd}(\sigma_1)}{\text{nrd}(I_s)} < n \right\} \approx \frac{6.45652}{2} \left(\frac{n}{\sqrt{p}} \right)^2 \quad (15)$$

Taking the desired probability to be 2^{-64} , which corresponds to NIST's recommended upper bound on the number of signatures, we obtain the following bound on the norm of the equivalent ideal, including the worst-case:

$$1 \leq 2^{-33} \sqrt{p} \leq \text{nrd}(I_r) \leq 2^{36} c_{\max}^2 \sqrt{p} \leq 8 \cdot c_{\max}^2 \cdot p, \quad (16)$$

since $\text{nrd}(I_s) \leq \text{nrd}(\sigma_1)$. For all the parameter sets of SQIsign, we have that $c_{\max} = 2^6$. Note that we do not need the heuristic that the LLL reduction is the same as the Minkowski reduction if we only work on $\mathcal{O}_0$ since we can efficiently compute it given Corollary 2.

Finally, we can go over the steps of Algorithm 3 to obtain the bound on the integers:

- **Line 2:** By Lemma 2, we have that for prime $\text{nrd}(I)$, the Minkowski reduction computation is upper-bounded by $8 \cdot \text{nrd}(I)^2 + p$, and $2^4 p \cdot \text{nrd}(I)^2$ otherwise. From [DLRW24, Lemma 48], the coefficients of $\alpha_t, \forall t \in \{1, 2, 3, 4\}$ are upper-bounded by $\sqrt{p \cdot \text{nrd}(I)}$ and $\sqrt{\text{nrd}(I)}$ for the basis elements $(1, i)$ and (j, k) respectively.

at each iteration one of the vectors decreases by a factor of $(1 + \eta)$, we can have at most $O(1 + \log \|\mathbf{b}_2\| - \log \lambda_1(L))$ iterations.

Let $\mathbf{b}_2^\perp = \mathbf{b}_2 - \mu\mathbf{b}_1$ be the orthogonal part of $\mathbf{b}_2$ and $\mathbf{r} = \mathbf{b}_2 - x\mathbf{b}_1$. At the last iteration of the first phase, we either terminate the algorithm or we have that

$$\|\mathbf{r}\| < \|\mathbf{b}_1\| \leq \|\mathbf{b}_2\| \leq (1 + \eta)\|\mathbf{r}\|. \quad (17)$$

Moreover, we have that $\|\mathbf{r}\|^2 = \|\mathbf{b}_2^\perp + (\mu - x)\mathbf{b}_1\|^2 \leq \|\mathbf{b}_2^\perp\|^2 + \|\mathbf{b}_1\|^2/2 \leq \|\mathbf{b}_2^\perp\|^2 + \|\mathbf{b}_2\|^2/2$. Combining this with Eq. (17), we obtain:

$$\frac{\|\mathbf{b}_2\|^2}{\|\mathbf{b}_2^\perp\|^2} \leq \left(\frac{1}{(1 + \eta)^2} - \frac{1}{2} \right)^{-1}, \quad (18)$$

where the upper bound is positive since $0 < \eta < \sqrt{2} - 1$. The above inequality implies that there is a finite number of vectors $\mathbf{w} \in L$ smaller than $\mathbf{b}_1$. This is shown in [NS04, Section 3.1] by writing $\mathbf{w} = w_1\mathbf{b}_1 + w_2\mathbf{b}_2$, $w_1, w_2 \in \mathbb{Z}[i]$. There, we have that $\|w_2\|\|\mathbf{b}_2^\perp\|^2 = \|\langle \mathbf{w}, \mathbf{b}_2^\perp \rangle\| \leq \|\mathbf{w}\|\|\mathbf{b}_2^\perp\| \leq \|\mathbf{b}_2\|^2$. Therefore,

$$\|w_2\| \leq \frac{\|\mathbf{b}_2\|^2}{\|\mathbf{b}_2^\perp\|^2} \leq \left(\frac{1}{(1 + \eta)^2} - \frac{1}{2} \right)^{-1}. \quad (19)$$

As such the number of possible values of w_2 is finite, regardless of the considered lattice. There are also finitely many possibilities for w_1 since $\|\mathbf{w}\|^2 = \|w_1 + \mu w_2\|^2 \|\mathbf{b}_1\|^2 + \|w_2\|^2 \|\mathbf{b}_2^\perp\|^2 \leq \|\mathbf{b}_1\|^2$. Because the algorithm is greedy, having finitely many vectors $\mathbf{w}$ regardless of the lattice implies that there are only $O(1)$ possible values that can be achieved after the first phase for any starting vectors. The overall number of iterations of the algorithm is thus $O(1 + \log \|\mathbf{b}_2\| - \log \lambda_1(L))$. $\square$

For stricter bounds, the analysis becomes more complicated. For instance, the local approach [NS04, Section 3.2] cannot be easily adapted because when $\|x\|^2 = 2$, we can have a contraction of the longer vector by a factor arbitrarily close to one. Now, from the above lemma, we have the following corollary:

Corollary 2. *We can efficiently compute the Minkowski-reduced basis $(\alpha_1, \alpha_2 = i\alpha_1, \alpha_3, \alpha_4 = i\alpha_3)$ of left $\mathcal{O}_0$ -ideals, where $\text{nrd}(\alpha_1) = \text{nrd}(\alpha_2)$ and $\text{nrd}(\alpha_3) = \text{nrd}(\alpha_4)$. A similar statement holds for right $\mathcal{O}_0$ -ideals.*

Note that the produced basis can be different from the one produced by the LLL algorithm. Moreover, a Minkowski-reduced basis is not unique. As such, replacing the lattice reduction algorithm leads to different signatures given the same randomness, thus failing to pass KAT file tests.

4 Application to SQIsign

Given the previous improvements, we revisit the analysis of SQIsign from [KLKL26] to obtain new integer bounds. We begin the analysis by first covering the main quaternion subroutines. We then track the size bounds when these subroutines are used in the key generation and signature generation of the SQIsign algorithm. Note that we do not cover the isogeny part since the operations there are performed over a field.

4.1 Analysis of quaternion subroutines

In this section, we analyze three algorithms used in SQIsign: the sampling of random ideals given a norm in Section 4.1.1, the quaternion part of the ideal to isogeny translation in Section 4.1.2 and finally the sampling of a random quaternion from an ideal given a bound in Section 4.1.3.

4.1.1 Random $\mathcal{O}_0$ -ideal given norm

Algorithm 5 RandomIdealGivenNorm $_{\mathcal{O}_0}(N)$ (simplified from [AAA⁺25])

Input: A positive integer N not a multiple of p which is the norm of some left $\mathcal{O}_0$ ideal.

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

- 1: Sample $\gamma \in \mathcal{O}_0$ and $\beta \in \mathcal{O}_0$ such that $\text{nrd}(\gamma) \equiv 0 \pmod{N}$ and $\gcd(\text{nrd}(\beta), N) = 1$.
 - 2: $J' \leftarrow$ the left $\mathcal{O}_0$ -ideal generated by $\gamma\beta$ and N
 - 3: **return** J'
-

Lemma 8. *The bound on the integer sizes during the computation of Algorithm 5 is $\max(4 \cdot N^2, 16p \cdot N)$.*

Proof. From [KLKL26, Lemma 8], γ and β can be generated with intermediate results below $16p \cdot N$. The resulting coefficients of the quaternions can be taken reduced modulo N , hence they are upper-bounded by N . From Corollary 1, the ideal creation can be performed with integers of size smaller than $4N^2$. $\square$

4.1.2 Ideal to Isogeny conversion

In the conversion of $\mathcal{O}_0$ -ideals to isogenies, the part that is computed over the quaternions is the sampling of ideals that satisfy a norm equation while the rest of the computation is performed in the isogeny domain. In order to do this, two main strategies are considered following the approach in [KLKL26]. The first one is an approach based on Clapotis [PR23]. This is the current approach used in SQIsign in which the sampling of the ideals is performed in the SuitableIdeals subroutine. The second approach is based on Qlapoti [BCRSE⁺26] which is more efficient than the previous approach. In the rest of the section, we revisit the analysis of [KLKL26] and obtain the following lemma:

Lemma 9. *Let I be a left $\mathcal{O}_0$ -ideal. Then, the quaternion part of IdealToIsogeny has intermediate integers bounded by $2^{10}p^2 \text{nrd}(I)^2$ if the SuitableIdeals subroutine is used. If Qlapoti is used, the upper bound on the integer size is $\max(2^{21}p^2, 8 \cdot \text{nrd}(I)^2 + p)$ if $\text{nrd}(I)$ is prime, and $16p \cdot \text{nrd}(I)^2$ otherwise.*

Proof. The proof is split into two parts, first we analyze the algorithm SuitableIdeals and then Qlapoti.

SuitableIdeals. From [KLKL26, Lemma 6], most of the operations can be upper-bounded by p^2 . The bottleneck during the running of the algorithm is the ideal multiplication between the ideal being converted I and the connecting ideals $\overline{J}_t$ and the Gram matrix formation of $\overline{J}_t I$. The ideal connects the $\mathcal{O}_0$ order to another maximal order $\mathcal{O}_t$. In the sampling algorithm, the ideal is used to connect I to another curve in order to increase the available randomness and thus make the failure probability of the algorithm less than 2^{-64} . Furthermore, given the precomputed values from the SQIsign implementation, the norms of the J_t are upper-bounded by $2^3 \sqrt{p}$.

Since both of the ideals are either right or left $\mathcal{O}_0$ -ideals, the multiplication can be upper-bounded by $16 \cdot \text{nrd}(\overline{J}_t)^2 \text{nrd}(I)^2 \leq 2^{10}p \cdot \text{nrd}(I)^2$ by Corollary 1. The resulting basis of the ideal product has elements upper-bounded by $2^5 \sqrt{p} \cdot \text{nrd}(I)$, hence the Gram matrix has elements at most the size $2^{10}p^2 \text{nrd}(I)^2$, which is the upper bound of the algorithm.

Qlapoti. We now analyze the bounds in every step of Algorithm 7.

- **Line 1:** Since the basis of I is in HNF and I is an $\mathcal{O}_0$ -ideal, we have by Lemma 1 that the coefficients of the basis are upper-bounded by $2 \cdot \text{nrd}(I)$. From Lemma 2, during the reduction the intermediate values are at most $16p \cdot \text{nrd}(I)^2$ or $8 \cdot \text{nrd}(I)^2 + p$,

depending on the primality of $\text{nrd}(I)$. Since the basis is Minkowski-reduced, the coefficients of the basis elements are upper-bounded by $\sqrt{p \cdot \text{nrd}(I)}$ and $\sqrt{\text{nrd}(I)}$ for the basis elements $(1, i)$ and (j, k) respectively ([DLRW24, Lemma 48]).

- **Line 2:** The intermediate values are upper-bounded by $4p \cdot \text{nrd}(I)$. Since the operation preserves the reduced property, the coefficients of the basis of the ideal J are upper-bounded by $\sqrt{p \cdot \text{nrd}(J)} \leq \sqrt{p \cdot \sqrt{p}}$. The latter inequality follows from [DLRW24, Lemma 12] since I is the shortest equivalent ideal.
- **Line 3:** From [DLRW24, Lemma 12], $\text{nrd}(J) \leq \sqrt{p}$.
- **Line 4:** From Lemma 10, α 's coefficients of the basis vectors $(1, i)$ and (j, k) are upper-bounded by $2^{19} \sqrt{p \cdot \sqrt{p}}$ and $2^{19} \sqrt[4]{p}$ respectively.
- **Line 5:** $\text{nrd}(\alpha) \leq 2^{38} p \cdot \sqrt{p}$, where the $\sqrt{p}$ comes from the norm of J , see Lemma 10. Therefore, by dividing it out we obtain $r \leq 2^{38} p$.
- **Line 7:** Since the computations are performed modulo n , the resulting coefficients s and t are bounded by $\sqrt{p}$.
- **Line 8:** z is upper-bounded by p since $2^e < p$ and s and t are upper-bounded by $\sqrt{p}$.
- **Lines 16, 18-19:** $z_1, z_2 \leq \sqrt{p}$ and $a_1, a_2 \leq \sqrt{p}$.
- **Line 20:** Assuming that $2^{19} < \sqrt[4]{p}$, the coefficients of β_1 and β_2 are upper-bounded by $2p$ for the coefficients of the quaternion basis $(1, i)$ and by $2^{19} \sqrt[4]{p}$ for (j, k) .
- **Line 21:** The intermediate computations are upper-bounded by the multiplication of the basis elements of J with β_1 and β_2 , the upper bound being in that case $2^{21} p^2$. Since the operation preserves the Minkowski reduction of J , the coefficients of I_1 are upper-bounded by $\sqrt{p \cdot \text{nrd}(I_1)} < p$ since $\text{nrd}(I_1) + \text{nrd}(I_2) = 2^e < p$. The same reasoning applies to I_2 .

The maximum integer size of Algorithm 7 is therefore $\max(2^{21} p^2, 8 \cdot \text{nrd}(I)^2 + p)$ if $\text{nrd}(I)$ is prime. Otherwise, the upper bound is $\max(2^{21} p^2, 16p \cdot \text{nrd}(I)^2)$, assuming that the upper bound of $\text{nrd}(I)$ is at least the norm of the output ideal in Algorithm 3. $\square$

Algorithm 6 SmallGenerator(I) [BCRSE⁺26, KLKL26]

Input: Left $\mathcal{O}_0$ -ideal I of norm N provided with a Minkowski-reduced basis $(\beta_1, \beta_2, \beta_3, \beta_4)$. Moreover, the ideal I is the shortest equivalent ideal.

Output: A small generator α of I .

- 1: Let $(\beta_1, \beta_2, \beta_3, \beta_4)$ be a Minkowski-reduced basis of I .
 - 2: Sample the coefficients uniformly (c_1, c_2, c_3, c_4) from $[-2^b, 2^b]$ for a bound $b = \text{QUAT\_qlapoti\_gen\_bound\_bits}$.
 - 3: $\alpha \leftarrow \sum_{i=1}^4 c_i \beta_i$ // $\alpha = a_\alpha + b_\alpha i + c_\alpha j + d_\alpha k$ for $a_\alpha, b_\alpha, c_\alpha, d_\alpha \in \frac{1}{2}\mathbb{Z}$
 - 4: **If** $\gcd(2a_\alpha, N) \neq 1$ or $\gcd(\text{nrd}(\alpha), N^2) \neq N$ **then goto line 2**
 - 5: **return** α
-

Algorithm 7 Qlapoti(I) [BCRSE⁺26]**Input:** Left $\mathcal{O}_0$ -ideal I with a basis in HNF.**Output:** Equivalent ideals I_1 and I_2 satisfying $\text{nrd}(I_1) + \text{nrd}(I_2) = 2^e$.

```

1: Let  $(\gamma_1, \gamma_2, \gamma_3, \gamma_4)$  be a Minkowski-reduced basis of  $I$ .
2:  $J \leftarrow \chi_I(\gamma_1)$  //  $J$  is the shortest equivalent ideal
3:  $n \leftarrow \text{nrd}(J)$ 
4:  $\alpha \leftarrow \text{SmallGenerator}(J)$  //  $\alpha = a_\alpha + b_\alpha i + c_\alpha j + d_\alpha k$  for  $a_\alpha, b_\alpha, c_\alpha, d_\alpha \in \frac{1}{2}\mathbb{Z}$ 
5:  $r \leftarrow \text{nrd}(\alpha)/n$ 
6: If  $\gcd(2a_\alpha, n) \neq 1$  then goto line 4
7: Compute shortest  $(s, t)$  such that  $2a_\alpha s + 2b_\alpha t = 2^e - 2r \pmod{n}$ , by solving a CVP problem.
8: Set  $z := 2(2^e - 2r - 2a_\alpha s - 2b_\alpha t)/n - s^2 - t^2$ 
9: If  $z < 0$  then goto line 4
10: If  $z \equiv 0 \pmod{4}$  and not  $s = t = 0 \pmod{2}$  then goto line 4
11: If  $z \equiv 1 \pmod{4}$  and not  $s \neq t \pmod{2}$  then goto line 4
12: If  $z \equiv 2 \pmod{4}$  and not  $s = t = 1 \pmod{2}$  then goto line 4
13: If  $z \equiv 3 \pmod{4}$  then goto line 4
14:  $\text{sol} \leftarrow \text{Cornacchia}(z)$ ; // May return  $\perp$  if no solution is found
15: If  $\text{sol} = \perp$  then goto line 4
16:  $z_1, z_2 \leftarrow \text{sol}$  //  $z_1^2 + z_2^2 = z$ 
17: If  $z_1 \not\equiv s \pmod{2}$  then swap  $z_1, z_2$ 
18:  $a_1 \leftarrow (z_1 + s)/2$ ,  $b_1 \leftarrow (z_2 + t)/2$ 
19:  $a_2 \leftarrow s - a_1$ ,  $b_2 \leftarrow t - b_1$ 
20:  $\beta_1 \leftarrow n(a_1 + b_1 i) + \alpha$ ,  $\beta_2 \leftarrow n(a_2 + b_2 i) + \alpha$ 
21:  $I_1 \leftarrow \chi_J(\beta_1)$ ,  $I_2 \leftarrow \chi_J(\beta_2)$ 
22: return  $I_1, I_2$ 

```

Lemma 10. *Algorithm 6 reaches at most an integer size of $2^{2b+4}p \cdot \sqrt{p}$, which equals $2^{38}p \cdot \sqrt{p}$ for $b = 17$ used by Qlapoti.*

Proof. We perform the analysis of Algorithm 6 line by line. In **line 1**, since the basis is Minkowski-reduced and I is the shortest ideal, the coefficients of the basis vectors 1 and i are upper-bounded by $\sqrt{p} \cdot \sqrt{p}$ and the others by $\sqrt[4]{p}$ ([DLRW24, Lemma 12, Lemma 48]). For **line 3**, we have that $|a_\alpha|, |b_\alpha| \leq 2^{b+2} \sqrt{p} \cdot \sqrt{p}$ and $|c_\alpha|, |d_\alpha| \leq 2^{b+2} \sqrt[4]{p}$. Finally, the reduced norm of α on **line 4** is upper-bounded by $2^{2b+4}p \cdot \sqrt{p}$. $\square$

4.1.3 Random Equivalent Quaternion

Algorithm 8 RandomEquivalentQuaternion(L) [AAA⁺25]**Input:** A lattice L forming the basis of $I_{\text{inter}} = \overline{I_{\text{com}}} \cap I_{sk} \cap I'_{chl}$.**Output:** A uniformly sampled $\mathbf{b} \in L$ such that $\text{nrd}(\mathbf{b}) < D_{rsp} \cdot D_{com} \cdot D_{sk} \cdot D_{chl} = B$, where $D_{com} = \text{nrd}(I_{com})$, $D_{sk} = \text{nrd}(I_{sk})$ and $D_{chl} = \text{nrd}(I'_{chl})$.

```

1: Let  $(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4)$  be a basis of  $L$ 
2: Compute the Gram matrix  $\mathbf{G}$  of  $(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4)$ 
3:  $\mathbf{U}, \mathbf{H} \leftarrow \text{L2}_{\delta, \eta}(\mathbf{Id}_d, \mathbf{G}^{-1})$  // Parse columns of  $\mathbf{U}$  and  $\mathbf{Id}_d$  as basis vectors
4: repeat
5:   for  $c$  from 1 to  $d$  do
6:      $B_c \leftarrow \lfloor \sqrt{B \mathbf{H}_{c,c}} \rfloor$ 
7:     Sample  $x_c$  uniformly from  $[-B_c, B_c]$ 
8:      $(y_1 \dots y_d) \leftarrow (x_1 \dots x_d) \mathbf{U}^{-1}$ 
9:   until  $(y_1 \dots y_d) \mathbf{G} (y_1 \dots y_d)^t \leq B$ 
10:  $\mathbf{b} \leftarrow y_1 \mathbf{b}_1 + \dots + y_d \mathbf{b}_d$ 
11: return  $\mathbf{b}$ 

```

Lemma 11. *Let $B' = \text{nrd}(\overline{I_{com}}) \cdot \text{nrd}(I_{sk}) \cdot \text{nrd}(I'_{chl})$. The integers in Algorithm 8 are upper-bounded by $2^{40}p^3 B'^2 B$.*

Proof. First, we observe that $B' \cdot \mathbb{Z}^4 \subset I_{\text{inter}}$ and the denominator is at most 2 since the

intersected ideals are all $\mathcal{O}_0$ -ideals² and have coprime norms. Furthermore, we assume that the ideal intersection was computed by Algorithm 2. We therefore have access to an upper-triangular HNF matrix $\mathbf{B}_{I_{inter}}^*$ with entries upper-bounded by $2 \cdot B'$. Moreover, we have that $\mathbf{B}_{I_{inter}}^\top = (\mathbf{B}_{I_{inter}}^*)^{-1}$ which implies by Lemma 3 that its entries are also upper-bounded by $4 \cdot B'$. On the other hand, the denominator of $\mathbf{B}_{I_{inter}}$ is 2 given that the intersection is over $\mathcal{O}_0$ -ideals. With that, we begin the analysis of the function line by line.

- **Line 2:** The entries of $\mathbf{G}$ are upper-bounded by $2^6 p \cdot B'^2$ since the matrix can be computed from $\mathbf{B}_{I_{inter}}$.
- **Line 3:** The entries of $\mathbf{G}^{-1}$ can be computed from $\mathbf{B}_{I_{inter}}^*$ and are thus upper-bounded by $2^4 p \cdot B'^2$ while the denominator is $p \cdot B'^2$. This can be reduced to $2^4 p \cdot B'$ and $p \cdot B'$ respectively since B' divides the reduced norm of the elements. The result follows from applying the Lemma 5 to the ideals $\overline{I_{com}}$ and $I_{sk} \cap I_{chl}$. From the lemma, we get that $p \text{ nrd}(\overline{I_{com}}) = p \cdot N_1$ and $p \text{ nrd}(I_{sk} \cap I_{chl}) = p \cdot N_2$ divide the reduced norms of the basis elements of $\overline{I_{com}}^*$ and $(I_{sk} \cap I_{chl})^*$ respectively. We also get that the basis elements of the dual ideals $\overline{I_{com}}^*$ and $(I_{sk} \cap I_{chl})^*$ admit a representation with the denominator $p \cdot N_1$ and $p \cdot N_2$ respectively. Let $\alpha^*, \beta^* \in \mathbb{Z}^4$ be the integer part of some elements in $\overline{I_{com}}^*$ and $(I_{sk} \cap I_{chl})^*$ respectively. Any element $\gamma \in \Lambda_{\overline{I_{com}} \cap I_{sk} \cap I_{chl}}^*$ can be written as³:

$$\gamma^* = \alpha^* / (p \cdot N_1) + \beta^* / (p \cdot N_2), \quad (20)$$

and the traces of two elements $\gamma_1^*, \gamma_2^* \in \Lambda_{\overline{I_{com}} \cap I_{sk} \cap I_{chl}}^*$:

$$p^2 N_1^2 N_2^2 \text{Tr}(\gamma_1^*, \gamma_2^*) = N_2^2 \text{Tr}(\alpha_1^*, \alpha_2^*) + N_1^2 \text{Tr}(\beta_1^*, \beta_2^*) \quad (21)$$

$$+ N_1 N_2 (\text{Tr}(\alpha_1^*, \beta_2^*) + \text{Tr}(\alpha_2^*, \beta_1^*)), \quad (22)$$

for some $\alpha_1^*, \alpha_2^* \in \overline{I_{com}}^*$ and $\beta_1^*, \beta_2^* \in (I_{sk} \cap I_{chl})^*$. This equation proves the statement.

For the L2 algorithm, we get from Lemma 12 that the intermediate results are bounded by $2^{16} p^2 B'^2$, where $m = 2 \cdot B'$ and $r = 2$.

Next we upper bound the norms of the output of the L2 algorithm. In particular we focus on the Eq. (2) where we bound the volume of $\Lambda_{I_{inter}}^*$. First, since we have that the Gram matrix of a dual lattice is the inverse of the Gram matrix of the lattice, we have that:

$$\text{Vol}(\Lambda_{I_{inter}}^*) \text{Vol}(\Lambda_{I_{inter}}) = 1 \quad (23)$$

Furthermore, for two lattices Λ and Λ' , if $\Lambda \subseteq \Lambda'$ then $\text{Vol}(\Lambda) = [\Lambda' : \Lambda] \text{Vol}(\Lambda')$, where $[\cdot : \cdot] \in \mathbb{N}_0$ is the index of Λ in Λ' . Given that $\Lambda_{I_{inter}} \subseteq \Lambda_{(I_{sk} \cap I_{chl})}$ and $\Lambda_{I_{inter}} \subseteq \Lambda_{\overline{I_{com}}}$ we have:

$$\text{Vol}(\Lambda_{I_{inter}}) = [\Lambda_{(I_{sk} \cap I_{chl})} : \Lambda_{I_{inter}}] \text{Vol}(\Lambda_{(I_{sk} \cap I_{chl})}) \quad (24)$$

$$= [\Lambda_{(I_{sk} \cap I_{chl})} : \Lambda_{I_{inter}}] \frac{p}{4} \text{nrd}(I_{sk} \cap I_{chl})^2, \text{ and} \quad (25)$$

$$\text{Vol}(\Lambda_{I_{inter}}) = [\Lambda_{\overline{I_{com}}} : \Lambda_{I_{inter}}] \text{Vol}(\Lambda_{\overline{I_{com}}}) \quad (26)$$

$$= [\Lambda_{\overline{I_{com}}} : \Lambda_{I_{inter}}] \frac{p}{4} \text{nrd}(\overline{I_{com}})^2, \quad (27)$$

²The result of the intersection is not, since $\overline{I_{com}}$ is a right $\mathcal{O}_0$ -ideal in contrast to the two others that are left- $\mathcal{O}_0$ -ideals. Hence, the LLL cannot be replaced by the Lagrange reduction algorithm.

³Note that we abuse notation by identifying quaternion elements with lattice elements.

where we use the property that the volume of an integral ideal I is equal to $\frac{p}{4} \text{nrd}(I)^2$. Because the two values are equal and the norms are coprime, we have that:

$$\text{Vol}(\Lambda_{I_{inter}}) \geq \frac{p}{4} \text{nrd}(\overline{I_{com}})^2 \text{nrd}(I_{sk} \cap I'_{chl})^2 = \frac{p}{4} B'^2. \quad (28)$$

As such, from Eqs. (2), (23) and (28), and by working over the integer part of the Gram matrix, we get that the norms of the (0.99, 0.501)-L2 reduced basis satisfy:

$$\prod_{i=1}^4 \text{nrd}(\alpha_i) \leq \left(2.48 \frac{\sqrt{(pB'^2)^4}}{pB'^2/4} \right)^2 \leq 2^7 p^2 B'^4. \quad (29)$$

where α_i is the integer part of $\alpha'_i \in I_{inter}^*$, $\forall i \in \{1, \dots, 4\}$ which is the basis output by the L2 algorithm. Since the norm of each element is divisible by B' , we get that $\text{nrd}(\alpha_i) \leq 2^7 p^2 B'$.

From this result we can deduce that the matrix $\mathbf{H}$, which is the Gram matrix of the basis α'_i , $\forall i \in \{1, \dots, 4\}$, has entries at most $2^7 p/B'$.

For $\mathbf{U}$, we look at the reduction as a lattice reduction of the dual $\mathbf{B}_{I_{inter}}^*$ which results in $\mathbf{B}_{I_{inter}}'^*$. We thus have that:

$$\mathbf{B}_{I_{inter}}'^* = \mathbf{B}_{I_{inter}}^* \mathbf{U} \implies (\mathbf{B}_{I_{inter}}^*)^{-1} \mathbf{B}_{I_{inter}}'^* = \mathbf{U} \quad (30)$$

We have that the entries of $\mathbf{B}_{I_{inter}}'^*$ are upper-bounded by $2^4 p \cdot \sqrt{B'}$ because of the bounds after the L2 reduction on the basis elements. Combining this with the upper bounds on the inverse of the dual basis, we get that the unimodular matrix elements are upper-bounded by $2^7 p \cdot \sqrt{B'}$, where a coefficient of $2 \cdot B'$ is simplified because of the determinant of both matrices.

- **Lines 6-7:** $B_c \leq 2^4 \sqrt{pD_{rsp}}$, $|x_c| \leq 2^4 \sqrt{pD_{rsp}}$.
- **Line 8:** When computing the inverse of the unimodular matrix, the elements of the adjugate can be upper-bounded by $6(2^7 p \cdot \sqrt{B'})^3 \leq 2^{24} (p \cdot \sqrt{B'})^3$ thus bounding the computation of the inverse.

To bound the result of the inversion, i.e. the entries of $\mathbf{U}^{-1}$, we follow the proof of [SD16, Lemma 10]. Concretely, any integer part γ of an element $\gamma' \in I_{inter}^*$ can be written as $\gamma = \sum_{i=1}^4 c_i \alpha_i$ or in the GSO basis:

$$\gamma = \sum_{i=1}^4 c_i \left(\alpha_i^\perp + \sum_{j=1}^{i-1} \mu_{i,j} \alpha_j^\perp \right) \quad (31)$$

for some $c_i \in \mathbb{Z}$. By taking the trace with each GSO vector, we get:

$$c_i \text{nrd}(\alpha_i^\perp) + \sum_{j=i+1}^4 c_j \mu_{j,i} \text{nrd}(\alpha_i^\perp) = \text{Tr}(\gamma, \alpha_i^\perp) \quad (32)$$

$$\iff |c_i| = |\text{Tr}(\gamma, \alpha_i^\perp) / \text{nrd}(\alpha_i^\perp) - \sum_{j=i+1}^4 c_j \mu_{j,i}| \quad (33)$$

$$\leq \sqrt{\text{nrd}(\gamma) / \text{nrd}(\alpha_i^\perp)} + \sum_{j=i+1}^4 |c_j| |\mu_{j,i}|, \quad (34)$$

where the last inequality follows from the Cauchy-Schwarz and triangle inequalities. From the properties of an LLL reduced basis, we have that $(\sqrt{\delta} - \eta)^{2(i-1)} \text{nrd}(\alpha_1) \leq \text{nrd}(\alpha_i^\perp)$ for all $i \in \{1, \dots, 4\}$. Hence, for all $i \in \{1, \dots, 4\}$ we get that:

$$|c_i| \leq 2^7 \sqrt{\text{nrd}(\gamma) / \text{nrd}(\alpha_1)} \quad (35)$$

Given that we want to upper bound the change of basis matrix from the reduced basis to the basis in HNF with vectors of norm at most $2^4 p \cdot B'^2$, we have that $|c_i| \leq 2^9 \sqrt{pB'}$, $\forall i \in \{1, \dots, 4\}$, which is also the upper-bound on the entries of $\mathbf{U}^{-1}$. As such, the coefficients $|y_i|$ are upper-bounded by $4 \cdot 2^4 \sqrt{pD_{rsp}} 2^9 \sqrt{pB'} = 2^{15} p \sqrt{B}$.

- **Line 9:** The intermediate computations during the vector-matrix multiplications are upper-bounded by $16 \cdot 2^6 p \cdot B'^2 \left(2^{15} p \sqrt{B}\right)^2 = 2^{40} p^3 B'^2 B$.
- **Line 10:** The intermediate values are upper-bounded by $4 \cdot 2^{15} p \sqrt{B} \cdot 4 \cdot B' = 2^{19} p \sqrt{B} B'$. On the other hand, the elements of the resulting quaternion $\mathbf{b}$ are upper-bounded by $\sqrt{B}$ since $\text{nrd}(\mathbf{b}) \leq B$.

□

4.2 SQIsign

As previously discussed, SQIsign is an isogeny-based signature scheme whose security is based on the difficulty of computing the endomorphism ring of a supersingular elliptic curve. The scheme is built from an underlying Σ -protocol, which is then made non-interactive using the Fiat-Shamir transform. We briefly recall the structure of this protocol, focusing on the quaternion operations that are relevant for our work. SQIsign is instantiated with primes $p \equiv 3 \pmod{4}$. In this setting, the curve $E_0 : y^2 = x^3 + x$ is supersingular and is used as a distinguished starting curve. This choice is convenient because its endomorphism ring $\text{End}(E_0)$ is explicitly known and under the Deuring correspondence, it can be identified with a special extremal maximal order $\mathcal{O}_0$ in the quaternion algebra $\mathcal{B}_{p,\infty}$.

During key generation as in Algorithm 9, SQIsign samples a random secret left $\mathcal{O}_0$ -ideal I_{sk} of prescribed large prime norm D_{mix} by calling `RandomIdealGivenNorm`. I_{sk} corresponds to the secret isogeny $\varphi_{sk} : E_0 \rightarrow E_{pk}$. Then I_{sk} is replaced by an equivalent ideal of smaller prime norm using `RandomEquivalentPrimeIdeal`. Under the Deuring correspondence, this step still preserves the corresponding target curve/order but provides for a smaller-norm representative for the next step of the key generation procedure. `IdealToIsogeny` then translates this ideal back into its corresponding isogeny evaluated on the required 2^f -torsion basis points. The public key thus contains the curve E_{pk} together with a hint for recovering a deterministic torsion basis, while the secret key stores the secret ideal I_{sk} and a change-of-basis matrix needed during signing.

The signing algorithm follows the usual commitment, challenge and response structure of the Fiat-Shamir-transformed Σ -protocol. From the point of view of quaternion arithmetic, the commitment phase is analogous to key generation as can be seen in Algorithm 10. The challenge is obtained by hashing the public key, the j -invariant of the commitment curve E_{com} , and the message. This challenge represents an isogeny originating from E_{pk} . In the quaternion domain, this challenge is converted into the corresponding challenge ideal I_{chl} through steps 11-13 as in Algorithm 10. At this point the signer has three relevant ideals: I_{sk}, I_{com}, I_{chl} . In the response phase, the signer samples a quaternion element α_{rsp} from $(\overline{I_{com}} \cap I_{sk} \cdot I_{chl})$ using `RandomEquivalentQuaternion`. The algorithm computes a basis and Gram matrix for this lattice representing $(\overline{I_{com}} \cap I_{sk} \cdot I_{chl})$ and samples a bounded-norm quaternion element from it using lattice sampling and lattice reduction. After this, SQIsign constructs a response ideal $I_{com,rsp} = \mathcal{O}_0 \alpha_{rsp} + \mathcal{O}_0(q_{rsp} D_{mix})$ using

the two-generator ideal representation discussed in Section 2. This ideal is handled through its underlying lattice representation and is eventually converted back into the response isogeny via `IdealToIsogeny` translation.

We now provide the bounds on integer sizes during key generation and the signing algorithms in the following two sub-sections.

4.2.1 Key generation

Algorithm 9 `SQIsign.KeyGen()` [AAA⁺25]

Output: Secret key `sk` and public key `pk`.

```

1: while true do
2:    $I_{sk} \leftarrow \text{RandomIdealGivenNorm}_{\mathcal{O}_0}(D_{mix})$  // Random ideal generation
3:   try
4:      $I_{sk} \leftarrow \text{RandomEquivalentPrimeIdeal}(I_{sk})$  // Smaller equivalent ideal sampling
5:      $E_{pk}, \varphi_{sk}(P_0), \varphi_{sk}(Q_0) \leftarrow \text{IdealToIsogeny}(I_{sk})$  // Computing  $E_{pk}$ 
6:   except
7:     continue
8:    $P_{pk}, Q_{pk}, \text{hint}_{pk} \leftarrow \text{TorsionBasisToHint}(E_{pk})$  // Precomputation of values used in signing
9:    $M_{sk} \leftarrow \text{ChangeOfBasis}_{2f}(E_{pk}, (\varphi_{sk}(P_0), \varphi_{sk}(Q_0)), (P_{pk}, Q_{pk}))$ 
10:   $pk \leftarrow (E_{pk}, \text{hint}_{pk})$ 
11:   $sk \leftarrow (E_{pk}, \text{hint}_{pk}, I_{sk}, M_{sk})$ 
12:  return  $sk, pk$ 

```

Theorem 1. *Algorithm 9 requires integers of size at most $2^{40}p^4$.*

Proof. In key generation, only lines 2, 4 and 5 are of interest for the bound analysis of quaternion operations since the remaining steps operate only in the isogeny domain. Hence we have that:

- **Line 2:** From Lemma 8, the computations are upper-bounded by $4D_{mix}^2$ and the resulting ideal I_{sk} has norm D_{mix} . In `SQIsign`, we have that $D_{mix} < 2^{16}p^2$.
- **Line 4:** From Lemma 6, the computations are upper-bounded by $2^{36}p^4$ since D_{mix} is prime. The resulting ideal I_{sk} has heuristically a norm bounded by $2^{48}\sqrt{p}$ and in the worst case by $2^{15} \cdot p$.
- **Line 5:** From Lemma 9, depending on whether the `IdealToIsogeny` subroutine uses a `SuitableIdeals`- or `Qlapoti`-based approach, we either have an upper bound of $2^{10}p^2 \text{nr}(I_{sk})^2$ or $\max(2^{21}p^2, 8 \cdot \text{nr}(I_{sk})^2 + p)$.

As such, the upper bound on the integers is heuristically $2^{36}p^4$, independently of the algorithms used. In the worst case, it becomes $2^{40}p^4$ if `SuitableIdeals` is used. For simplicity, we consider only the worst-case bound. $\square$

4.2.2 Signature generation

Algorithm 10 SQIsign.Sign(**sk**, **msg**) adapted from [AAA⁺25]

Input: Secret signing key **sk** and message **msg** $\in \{0, 1\}^*$.

Output: Signature σ .

```

1: Parse sk as  $(E_{pk}, \text{hint}_{pk}, I_{sk}, \mathbf{M}_{sk})$ 
2:  $P_{pk}, Q_{pk} \leftarrow \text{TorsionBasisFromHint}(E_{pk}, \text{hint}_{pk})$ 
3: while true do
4:    $I_{com} \leftarrow \text{RandomIdealGivenNorm}_{\mathcal{O}_0}(D_{mix})$ 
5:   try
6:      $I_{com} \leftarrow \text{RandomEquivalentPrimeIdeal}(I_{com})$ 
7:      $E_{com}, P_{com}, Q_{com} \leftarrow \text{IdealToIsogeny}(I_{com})$ 
8:   except
9:     continue
10:   $\text{chl} \leftarrow \text{HASH}(\text{pk} \parallel j(E_{com}) \parallel \text{msg})$ 
11:   $(c_1, c_2) \leftarrow \mathbf{M}_{sk} \cdot (1, \text{chl})$ 
12:   $I'_{chl} \leftarrow \text{KernelDecomposedToIdeal}_{2^f}(c_1, c_2)$ 
13:   $I_{sk} \cdot I_{chl} \leftarrow I_{sk} \cap I'_{chl}$ 
14:   $\alpha_{rsp} \leftarrow \text{RandomEquivalentQuaternion}(\overline{I_{com} \cap (I_{sk} \cdot I_{chl})})$ 
15:   $\alpha_{rsp}, n_{bt} \leftarrow \text{ComputeBacktrackingAndNormalize}(\alpha_{rsp})$ 
16:   $d_{rsp} \leftarrow \text{nrd}(\alpha_{rsp}) / (\text{nrd}(I_{com}) \cdot \text{nrd}(I_{sk}) \cdot 2^f) \parallel \text{nrd}(I_{chl}) = 2^f$ 
17:   $r_{rsp} \leftarrow \text{DyadicValuation}(d_{rsp})$ 
18:   $q_{rsp} \leftarrow d_{rsp} / 2^{r_{rsp}}$ 
19:   $I_{com, rsp} \leftarrow \mathcal{O}_0 \alpha_{rsp} + \mathcal{O}_0(q_{rsp} \text{nrd}(I_{com}))$ 
20:   $e'_{rsp} \leftarrow e_{rsp} - r_{rsp} - n_{bt}$ 
21:  if  $e'_{rsp} > 0$  then
22:    try
23:       $I_{aux} \leftarrow \text{RandomIdealGivenNorm}_{\mathcal{O}_0}(2^{e'_{rsp}} - q_{rsp})$ 
24:       $E'_{aux}, P'_{aux}, Q'_{aux} \leftarrow \text{IdealToIsogeny}(I_{com, rsp} \cap I_{aux})$ 
25:    except
26:      continue
27:     $(E_{aux}, P_{aux}, Q_{aux}, E_{chl}, P_{chl}, Q_{chl}) \leftarrow \text{SplitAuxiliaryIsogeny}$ 
28:     $(E_{com}, E'_{aux}, P_{com}, Q_{com}, P'_{aux}, Q'_{aux}, q_{rsp}, e'_{rsp}, r_{rsp})$ 
29:  else
30:    try
31:       $E_{chl}, P_{chl}, Q_{chl} \leftarrow \text{IdealToIsogeny}(I_{com, rsp})$ 
32:       $E_{aux}, P_{aux}, Q_{aux} \leftarrow E_{chl}, P_{chl}, Q_{chl}$ 
33:    except
34:      continue
35:  if  $r_{rsp} > 0$  then
36:     $E_{chl}, P_{chl}, Q_{chl} \leftarrow \text{ComputeEvenNonBacktrackingResponse}(E_{chl}, P_{chl}, Q_{chl}, \alpha_{rsp}, e'_{rsp}, r_{rsp})$ 
37:   $E_{chl}, P_{chl}, Q_{chl} \leftarrow \text{ComputeChallengeIsogeny}(E_{pk}, \text{chl}, P_{pk}, Q_{pk}, E_{chl}, P_{chl}, Q_{chl}, n_{bt})$ 
38:   $\mathbf{M}_{chl}, \text{hint}_{aux}, \text{hint}_{chl} \leftarrow \text{SetChangeOfBasisMatrix}(E_{aux}, E_{chl}, P_{aux}, Q_{aux}, P_{chl}, Q_{chl}, e'_{rsp} + r_{rsp})$ 
39:  return  $\sigma$ 

```

Theorem 2. *The integers in Algorithm 10 are upper-bounded heuristically by $2^{328} \sqrt{pp}^9$ and in the worst case by $2^{130} \sqrt{pp}^{12}$. Moreover, if the subroutine *RandomEquivalentQuaternion* is not considered, the upper bounds become $2^{110} p^5$ and $2^{62} p^6$ respectively.*

Proof. For the analysis of the signing algorithm, we focus only on the operations over the quaternions. The analysis is as follows:

- **Lines 4-7:** The same analysis as in Algorithm 9 applies. Therefore, by Theorem 1, the bound of the computation is $2^{40} p^4$. Moreover, we have that the norm of the commitment ideal is heuristically bounded by $2^{48} \sqrt{p}$ and in the worst-case by $2^{15} \cdot p$.
- **Line 12:** In the subroutine *KernelDecomposedToIdeal*, the only quaternion computation is the creation of an ideal of norm $2^f \leq p$ [AAA⁺25, Algorithm 3.17], hence the computation is upper-bounded by the HNF computation which requires an integer size of at most $4p^2$.

- **Lines 13-14:** Let $B' = \text{nrd}(I_{com}) \cdot \text{nrd}(I_{sk}) \cdot \text{nrd}(I'_{chl})$ which is upper-bounded by either $2^{96}p^2$ or $2^{30}p^3$, depending on whether or not the heuristic is applied to the norms of I_{sk} and I_{com} . The computation of the intersection of the ideals can be performed with integer sizes of at most $4B'^2$ according to Lemma 4. From Lemma 11, the subroutine uses integers of size at most $2^{40} \cdot D_{rsp}p^3B'^3$. The resulting quaternion α has a reduced norm bounded by $\text{nrd}(\alpha) \leq D_{rsp}B'$. Therefore, the heuristic bounds are $2^{328}\sqrt{pp}^9$ and $2^{97}\sqrt{pp}^2$ for the intermediate computations and $\text{nrd}(\alpha)$ respectively. The worst-case bounds are $2^{130}\sqrt{pp}^{12}$ and $2^{31}\sqrt{pp}^3$ respectively.
- **Line 15:** By [KLKL26, Lemma 26], the integers are at most as large as those in α , which is at most $2^{49}\sqrt[4]{pp}$ and $2^{16}\sqrt{\sqrt{pp}^3}$ for the heuristic case and the worst case respectively.
- **Line 16:** Since $\text{nrd}(\alpha) \leq D_{rsp} \cdot B'$, d_{rsp} is at most $D_{rsp} < 2\sqrt{p}$.
- **Line 19:** The creation of the ideal requires at most $4 \cdot (2\sqrt{p} \cdot \text{nrd}(I_{com}))^2$, which heuristically is upper-bounded by $2^{100}p^2$ and at worst by $2^{34} \cdot p^3$.
- **Line 23:** From Lemma 8, the computations are upper-bounded by $2^{e'_{rsp}+4}p \leq 2^5\sqrt{pp}$ and the resulting ideal I_{aux} has norm at most $2\sqrt{p}$.
- **Line 24:** From Lemma 9, depending on whether the IdealToIsogeny subroutine uses SuitableIdeals or Qlapoti based approach, we either have an upper bound of $2^{10}p^2 \cdot \text{nrd}(I_{com,rsp})^2 \text{nrd}(I_{aux})^2$ or $2^4p \cdot \text{nrd}(I_{com,rsp})^2 \text{nrd}(I_{aux})^2$. We have heuristic upper bounds of $2^{110}p^5$ and $2^{104}p^4$ respectively and worst-case bounds of $2^{44}p^6$ and $2^{38}p^5$ respectively.
- **Line 30:** The same analysis applies as in **Line 24**, but with $\text{nrd}(I_{aux}) = 1$.
- **Line 35:** By [KLKL26, Lemma 27], the sizes of the integers are at most $\text{nrd}(\alpha)$.

□

4.3 Summary and discussion

Our derived bounds are summarized in Table 1 and compared to the previous works. For the key generation algorithm, using our methods, the obtained bound is less than the square root of the results from [KLKL26] and we match [KLY26] up to a constant factor. Similarly, for the overall signing computation, we also obtain a considerable improvement compared to [KLKL26]. However, compared to [KLY26], we achieve a higher upper bound. This is due to the different assumptions made on the unimodular matrix $\mathbf{U}$ in Algorithm 8. Unlike ours, the previous analyses assume that the matrix $\mathbf{U}$ has entries at most equal to 1 [KLKL26, Lemma 25]. Because of that, they do not observe the same blowup as in our case. If we consider the results without Algorithm 8, our bounds become better for the remaining parts of the signing in both cases. In practice, we observe experimentally that the upper bound is $O(p^7)$. We observe that this reduction comes from two contributions in Algorithm 8: the integer sizes of the norms of the L2 reduced basis being of similar sizes, contributing to a reduction of $O(p^{1.5})$, and the norms of the HNF basis being smaller by $O(p)$. This indicates that there is potentially room to make the bounds tighter.

Table 1: Comparison of integer bounds in SQIsign key and signature generation.

	KeyGen		SQIsign.Sign		SQIsign.Sign ¹	
	Heuristic	Worst	Heuristic	Worst	Heuristic	Worst
[KLKL26]	-	$2^{68}p^9$	-	$2^{16}p^{28}$	-	$2^{16}p^{28}$
[KLY26] ²	$2^{24}p^4$	-	$2^{21}p^7$	-	$2^{21}p^7$	-
This work	$2^{40}p^4$	$2^{40}p^4$	$2^{328}\sqrt{pp}^9$	$2^{130}\sqrt{pp}^{12}$	$2^{110}p^5$	$2^{62}p^6$

¹ Upper bounds of the signing operation where computations involving $\mathbf{U}^{-1}$ from Algorithm 8 are not considered.

² The HNF computations are replaced by lattice reductions over lattices in $\mathbb{Z}^{4 \times n}$, with $n \in \{4, 8, 16\}$.

5 Conclusion and future works

In this work, we revisited the integer-size analysis of the quaternion arithmetic layer of SQIsign. Exploiting the containment property $\text{nrd}(I) \cdot \mathbb{Z}^4 \subset \Lambda_I$ satisfied by all $\mathcal{O}_0$ -ideals in the scheme, we halved the bit length of the modulus in the modular HNF computation, obtained an exact-division matrix inversion algorithm with linearly bounded intermediate values, and justified the upper-triangular HNF convention used in the reference implementation. We further showed that $\mathcal{O}_0$ -ideals can be reduced as two-dimensional lattices over the Gaussian integers, replacing LLL by Lagrange reduction, an algorithm that requires no floating-point arithmetic and provably outputs a Minkowski-reduced basis, making it well suited to constant-time implementations. As a result, our key generation and signing bounds improve on [KLKL26] by more than a square root. Compared to [KLY26], we match the key generation upper bound up to a constant factor and we improve the signing upper bound if we do not consider the quaternion response sampling.

For future work, the main open problem is tightening the bounds on the unimodular matrix $\mathbf{U}$ and its inverse in Algorithm 8, which dominate our worst-case signing bound. In practice, we encounter matrices that are smaller than our analysis suggests, on the order of $O(p^7)$ instead of $O(p^{9.5})$. This would require tightening the bounds on the HNF representation of a random ideal and the sizes of L2 reduced vectors. Another direction is to derive an explicit worst-case iteration bound on the Lagrange reduction over $\mathbb{Z}[i]$. Finally, our techniques should apply to other schemes built on the Deuring correspondence, such as PRISM [BBC⁺25].

Acknowledgements

Maciej Czaprynko and Anisha Mukherjee are supported by the Austrian FWF grant PAT6402023, and Sujoy Sinha Roy is partially supported by the State Government of Styria, Austria, through the Department Zukunftsfonds Steiermark.

AI acknowledgment. During this work, Claude AI was used for improving the text and for literature surveys.

A Integer bounds on the L2 algorithm

In this section, we analyze the integer bounds of the L2 algorithm defined in Algorithm 12. We focus on the inputs of the algorithm as in Algorithm 8 and thus we adapt the definition of the L2 from [NS09] to accept a set of vectors $(\mathbf{u}_1, \dots, \mathbf{u}_r)$ that does not need to be related to the Gram matrix $\mathbf{G}$ of a basis $(\mathbf{b}_1, \dots, \mathbf{b}_r)$. In particular, by setting $\mathbf{u}_i = \mathbf{b}_i, \forall i \in \{1, \dots, r\}$ we recover the base definition. On the other hand, setting $\mathbf{u}_i = \mathbf{e}_i$, where $\mathbf{e}_i$ is a vector of zeros except at the i -th position where it is equal to 1, the resulting vector defines the change of basis matrix $\mathbf{U}$ between $(\mathbf{b}_1, \dots, \mathbf{b}_r)$ and the LLL-reduced vectors. The latter setting is the one used in SQIsign.

Algorithm 11 Lazy Size-Reduction Algorithm, based on [NS09]

Input: A factor $\eta > 1/2$, an fp-precision $\ell + 1$, an index κ , a set of vectors $(\mathbf{u}_1, \dots, \mathbf{u}_r)$, a Gram matrix $\mathbf{G}$ of a basis $(\mathbf{b}_1, \dots, \mathbf{b}_r)$ basis, and floating-point numbers $\bar{r}_{i,j}$ and $\bar{\mu}_{i,j}$ for $j \leq i < \kappa$.

Output: Floating-point numbers $\bar{r}_{\kappa,j}$, $\bar{\mu}_{\kappa,j}$, and $\bar{s}_j^{(\kappa)}$ for $j \leq \kappa$, vectors $(\mathbf{u}_1, \dots, \mathbf{u}_{\kappa-1}, \mathbf{u}'_{\kappa}, \mathbf{u}_{\kappa+1}, \dots, \mathbf{u}_d)$, where $\mathbf{u}'_{\kappa} = \mathbf{u}_{\kappa} - \sum_{i < \kappa} x_i \cdot \mathbf{u}_i$ for some $x_i \in \mathbb{Z}$, and the Gram matrix of the basis $(\mathbf{b}_1, \dots, \mathbf{b}_{\kappa-1}, \mathbf{b}'_{\kappa}, \mathbf{b}_{\kappa+1}, \dots, \mathbf{b}_d)$, where $\mathbf{b}'_{\kappa} = \mathbf{b}_{\kappa} - \sum_{i < \kappa} x_i \cdot \mathbf{b}_i$ and $|\langle \mathbf{b}'_{\kappa}, \mathbf{b}_i^{\perp} \rangle| \leq \eta \|\mathbf{b}_i^{\perp}\|^2$ for any $i < \kappa$.

- 1: $\bar{\eta} \leftarrow \frac{\diamond(\diamond(\eta) + 1/2)}{2}$
 - 2: Compute the $\bar{r}_{\kappa,j}$'s, $\bar{\mu}_{\kappa,j}$'s, $\bar{s}_j^{(\kappa)}$'s using floating-point Cholsky's factorization [NS09].
 - 3: **if** $\max_{j < \kappa} |\bar{\mu}_{\kappa,j}| \leq \bar{\eta}$, **terminate**. **Else**, **for** $i = \kappa - 1$ **down to** 1 **do**
 - 4: $X_i \leftarrow \lfloor \bar{\mu}_{\kappa,i} \rfloor$
 - 5: **for** $j = 1$ **to** $i - 1$, $\bar{\mu}_{\kappa,j} \leftarrow \diamond(\bar{\mu}_{\kappa,j} - \diamond(X_i \cdot \bar{\mu}_{i,j}))$
 - 6: $\mathbf{u}_{\kappa} \leftarrow \mathbf{u}_{\kappa} - \sum_{i=1}^{\kappa-1} X_i \cdot \mathbf{u}_i$, update $\mathbf{G}$ accordingly
 - 7: **Goto** step 2
-

Algorithm 12 L2 $_{\delta,\eta}(\mathbf{U}, \mathbf{G})$, based on [NS09]

Input: A pair (δ, η) , with $\delta \in (1/4, 1)$ and $(1/2, \sqrt{\delta})$, a set of vectors $(\mathbf{u}_1, \dots, \mathbf{u}_r)$ contained in $\mathbf{U}$, a Gram matrix $\mathbf{G}$ of a basis $(\mathbf{b}_1, \dots, \mathbf{b}_r)$, and an fp-precision $\ell + 1$.

Output: A (δ, η) -LLL reduced basis.

- 1: $\bar{\delta} \leftarrow \frac{\diamond(\diamond(\delta) + 1)}{2}$, $\bar{r}_{1,1} \leftarrow \diamond(\langle \mathbf{b}_1, \mathbf{b}_1 \rangle)$, $\kappa \leftarrow 2$
 - 2: **while** $\kappa \leq d$ **do**
 - 3: Size-reduce the vector $\mathbf{u}_{\kappa}$ using the Algorithm 11, updating the floating-point GSO.
 - 4: $\kappa' \leftarrow \kappa$
 - 5: **while** $\kappa \geq 2$ **and** $\diamond(\bar{\delta} \cdot \bar{r}_{\kappa-1, \kappa-1}) > \bar{s}_{\kappa-1}^{(\kappa')}$ **do**
 - 6: $\kappa \leftarrow \kappa - 1$
 - 7: **for** $i = 1$ **to** $\kappa - 1$ **do**
 - 8: $\bar{\mu}_{\kappa,i} \leftarrow \bar{\mu}_{\kappa',i}$, $\bar{r}_{\kappa,i} \leftarrow \bar{r}_{\kappa',i}$
 - 9: $\bar{r}_{\kappa,\kappa} \leftarrow \bar{s}_{\kappa}^{(\kappa')}$
 - 10: Insert the vector $\mathbf{u}_{\kappa'}$ right before the vector $\mathbf{u}_{\kappa}$ and update $\mathbf{G}$ accordingly
 - 11: $\kappa \leftarrow \kappa + 1$
 - 12: **return** $(\mathbf{u}_1, \dots, \mathbf{u}_r)$
-

Lemma 12. *Let $(\mathbf{b}_1, \dots, \mathbf{b}_4) \in \mathbb{Z}^{4 \times 4}$ be an HNF basis of a lattice Λ that satisfies $m\mathbb{Z}^4 \subset \Lambda$ for some $m \in \mathbb{N}$. Furthermore, let $r \in \mathbb{N}m$, with $r \leq m$, be an integer such that m/r divides the Gram matrix $\mathbf{G}$ defined by the basis. With inputs the identity matrix $\mathbf{Id}_4$ and the Gram matrix $\mathbf{G}$ divided by m/r , the intermediate integer sizes during the computation of $\text{L2}_{\delta,\eta}(\mathbf{Id}_4, r\mathbf{G}/m)$, with $\delta = 0.99$ and $\eta = 0.501$, are upper-bounded by $2^{13}r \cdot p^2m^2$.*

Proof. For the proof, we consider the fact that the usual Euclidean norm and scalar product can be replaced by the equivalent notions in the quaternion algebra.

For the integer bounds, every step of the computation either involves only floating-point operations, the size of which is bounded by the precision ℓ , or swapping operations. The former is analyzed in [NS09] and set in the SQIsign implementation while the latter does

not modify the values of any element. The sole exception is Line 6 in Algorithm 11 which updates the vector $\mathbf{u}_\kappa$ and the Gram matrix, which is the focus of the remainder of the proof.

From [NS09, Theorem 6], [NS09, Lemma 7] and Lemma 1 we have for all $i \in \{1, \dots, 4\}$ that:

$$\text{nrd}(\mathbf{b}_i^{(t)}) \leq 4 \max_{j \in \{1, \dots, 4\}} \text{nrd}(\mathbf{b}_j^{(1)}) \leq 4p \cdot m^2, \text{ and} \quad (36)$$

$$|X_i^{(t)}| \leq \eta + 1 + 4(\delta - \eta^2)^{-2} \sqrt{p \cdot m} \leq 8\sqrt{p \cdot m}, \quad (37)$$

where $\mathbf{b}_i^{(t)}$ and $X_i^{(t)}$ are the variables $\mathbf{b}_i$ and X_i defined in Algorithm 12 at the t -th iteration of the outer loop. We also assume that $\sqrt{p \cdot m} > 2$. The first equation implies that at each iteration the absolute values of coefficients of $\mathbf{b}_i$ are upper-bounded by $2\sqrt{p} \cdot m$ and $2m$ for the basis elements corresponding to $(1, i)$ and (j, k) respectively. Furthermore, the values in the Gram matrix are upper-bounded by $4p \cdot m$.

Given the inputs to the L2 algorithm, at each iteration t the vectors $(\mathbf{u}_1^{(t)}, \dots, \mathbf{u}_4^{(t)})$ represent the columns of a unimodular matrix $\mathbf{U}^{(t)}$, such that $\mathbf{B}^{(t)} = \mathbf{B}^{(1)} \mathbf{U}^{(t)}$ where the matrices $\mathbf{B}^{(t)}$ and $\mathbf{B}^{(1)}$ correspond to the basis elements $\mathbf{b}_i^{(t)}$ and $\mathbf{b}_i^{(1)}$ respectively. As such, $\mathbf{U}^{(t)} = (\mathbf{B}^{(1)})^{-1} \mathbf{B}^{(t)}$. From Lemma 3, the matrix can be computed using integers with size at most $16\sqrt{p} \cdot m^2$ and the resulting matrix has coefficients at most $16\sqrt{p} \cdot m$ since it is unimodular and thus can be divided by the denominator of $(\mathbf{B}^{(1)})^{-1}$ which is m .

The remaining parts are the update of $\mathbf{u}_\kappa^{(t)}$ and the corresponding entries in the Gram matrix $\mathbf{G}$. In the first case, we have that, for all $\kappa \in \{1, \dots, 4\}$ and $j \in \{1, \dots, 4\}$,

$$|\mathbf{u}_\kappa^{(t)}[j] - \sum_{i=1}^{\kappa-1} X_i \cdot \mathbf{u}_i[j]| \leq |\mathbf{u}_\kappa^{(t)}[j]| + \sum_{i=1}^3 |X_i \cdot \mathbf{u}_i[j]| \leq 2^8 p \cdot m \sqrt{m} \quad (38)$$

using the triangle inequality and from the fact that $\kappa \leq 4$. For the Gram matrix, the limiting factor is the norm update and thus we omit the update of the scalar products. In that case, we have that the reduced norm at iteration t is computed as:

$$\text{nrd}\left(\sum_{i=1}^{\kappa} X_i \cdot \mathbf{b}_i^{(t)}\right) = \sum_{i=1}^{\kappa} X_i^2 \text{nrd}(\mathbf{b}_i^{(t)}) + \sum_{i=1}^{\kappa} \sum_{j=1}^{\kappa} X_i X_j \text{Tr}(\mathbf{b}_i^{(t)}, \mathbf{b}_j^{(t)}) \quad (39)$$

$$\leq \sum_{i=1}^4 X_i^2 \text{nrd}(\mathbf{b}_i^{(t)}) + \sum_{i=1}^4 \sum_{j=1}^4 X_i X_j \sqrt{\text{nrd}(\mathbf{b}_i^{(t)}) \text{nrd}(\mathbf{b}_j^{(t)})} \quad (40)$$

$$\leq 2^{13} p^2 m^3 \quad (41)$$

where we define $X_\kappa = -1$ and $X_i = 0, \forall i \in \kappa + 1, \dots, 4$. Assuming that the input Gram matrix was divided by m/r , we have that the intermediate computation can be upper-bounded by $2^{13} r \cdot p^2 m^2$, which corresponds to the overall upper bound. $\square$

References

- [AAA⁺25] Marius A. Aardal, Gora Adj, Diego F. Aranha, Andrea Basso, Isaac Andrés Canales Martínez, Jorge Chávez-Saab, Maria Corte-Real Santos, Pierrick Dartois, Luca De Feo, Max Duparc, Jonathan Komada Eriksen, Tako Boris Fouotsa, Décio Luiz Gazzoni Filho, Basil Hess, David Kohel, Antonin Leroux, Patrick Longa, Luciano Maino, Michael Meyer, Kohei Nakagawa, Hiroshi Onuki, Lorenz Panny, Sikhar Patranabis, Christophe Petit, Giacomo Pope, Krijn Reijnders, Damien Robert, Francisco Rodríguez-Henríquez, Sina Schaeffler, and Benjamin Wesolowski. SQIsign. Technical report, National Institute of Standards and Technology, 2025.

- [AAA⁺26] Marius A. Aardal, Gora Adj, Diego F. Aranha, Andrea Basso, Giacomo Borin, Isaac Andrés Canales Martínez, Jorge Chávez-Saab, Maria Corte-Real Santos, Pierrick Dartois, Luca De Feo, Max Duparc, Thomas Espitau, Jonathan Komada Eriksen, Tako Boris Fouotsa, Décio Luiz Gazzoni Filho, Basil Hess, Riccardo Invernizzi, David Kohel, Antonin Leroux, Patrick Longa, Luciano Maino, Michael Meyer, Marzio Mula, Kohei Nakagawa, Hiroshi Onuki, Lorenz Panny, Sikhar Patranabis, Christophe Petit, Giacomo Pope, Krijn Reijnders, Damien Robert, Francisco Rodríguez-Henríquez, Sina Schaeffler, Frederik Vercauteren, Alexandre Wallet, Benjamin Wesolowski, and Wessel van Woerden. SQIsign. Technical report, National Institute of Standards and Technology, 2026.
- [ABC⁺26] Gorjan Alagic, Maxime Bros, Pierre Ciadoux, Quynh Dang, Thinh Dang, John Kelsey, Jacob Lichtinger, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Hamilton Silberg, Daniel Smith-Tone, and Noah Waller. Status report on the second round of the additional digital signature schemes for the NIST post-quantum cryptography standardization process. National Institute of Standards and Technology, 2026.
- [BBC⁺25] Andrea Basso, Giacomo Borin, Wouter Castryck, Maria Corte-Real Santos, Riccardo Invernizzi, Antonin Leroux, Luciano Maino, Frederik Vercauteren, and Benjamin Wesolowski. PRISM: simple and compact identification and signatures from large prime degree isogenies. In Tibor Jager and Jiaxin Pan, editors, *Public-Key Cryptography - PKC 2025 - 28th IACR International Conference on Practice and Theory of Public-Key Cryptography, Røros, Norway, May 12-15, 2025, Proceedings, Part III*, volume 15676 of *Lecture Notes in Computer Science*, pages 300–332. Springer, 2025.
- [BCRSE⁺26] Giacomo Borin, Maria Corte-Real Santos, Jonathan Komada Eriksen, Riccardo Invernizzi, Marzio Mula, Sina Schaeffler, and Frederik Vercauteren. Qlapoti: Simple and efficient translation of quaternion ideals to isogenies. In Goichiro Hanaoka and Bo-Yin Yang, editors, *Advances in Cryptology – ASIACRYPT 2025*, pages 174–205, Singapore, 2026. Springer Nature Singapore.
- [BHJ⁺25] Andrea Basso, Chenfeng He, David Jacquemin, Fatna Kouider, Péter Kutas, Anisha Mukherjee, Sina Schaeffler, and Sujoy Sinha Roy. Constant-time quaternion algorithms for SQIsign. Cryptology ePrint Archive, Paper 2025/2192, 2025.
- [Cle25] James Clements. Structural results for maximal quaternion orders and connecting ideals of prime power norm in $b_{p,\infty}$. Cryptology ePrint Archive, Paper 2025/042, 2025.
- [Coh93] Henri Cohen. *A course in computational algebraic number theory*. Springer Science & Business Media, 1993.
- [Deu41] Max Deuring. Die typen der multiplikatorenringe elliptischer funktionenkörper. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 14:197–272, 1941.
- [DLRW24] Pierrick Dartois, Antonin Leroux, Damien Robert, and Benjamin Wesolowski. Sqsignhd: new dimensions in cryptography. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 3–32. Springer, 2024.

- [ESWvW] Thomas Espitau, Sina Schaeffler, Alexandre Wallet, and Wessel van Woerden. Lattice reduction algorithms for sqsign. to appear. 2026.
- [HKK⁺25] Ottó Hanyecz, Alexander Karenin, Elena Kirshanova, Péter Kutas, and Sina Schaeffler. Constant time lattice reduction in dimension 4 with application to sqsign. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2025(2):511–534, 2025.
- [KLKL26] Won Kim, Jeonghwan Lee, Hyeonhak Kim, and Changmin Lee. Sqsign with fixed-precision integer arithmetic. In Shi Bai and Edoardo Persichetti, editors, *Public-Key Cryptography - PKC 2026 - 29th IACR International Conference on Practice and Theory of Public-Key Cryptography, West Palm Beach, FL, USA, May 25-28, 2026, Proceedings, Part III*, volume 16553 of *Lecture Notes in Computer Science*, pages 3–30. Springer, 2026.
- [KLY26] Won Kim, Changmin Lee, and Hyunwoo Yoo. Compact quaternion algorithms for SQIsign. Cryptology ePrint Archive, Paper 2026/1031, 2026.
- [KMJK25] Fatna Kouider, Anisha Mukherjee, David Jacquemin, and Péter Kutas. Constant-time integer arithmetic for sqsign. In Abderrahmane Nitaj, Svetla Petkova-Nikova, and Vincent Rijmen, editors, *Progress in Cryptology - AFRICACRYPT 2025 - 16th International Conference on Cryptology in Africa, Rabat, Morocco, July 21-23, 2025, Proceedings*, volume 15651 of *Lecture Notes in Computer Science*, pages 192–215. Springer, 2025.
- [Ler26] Antonin Leroux. EFFICIENT QUATERNION ALGORITHMS FOR THE DEURING CORRESPONDENCE, AND APPLICATION TO THE EVALUATION OF MODULAR POLYNOMIALS. Cryptology ePrint Archive, Paper 2026/185, 2026.
- [LLL82] Arjen K. Lenstra, Hendrik W. Lenstra, and László Miklós Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982.
- [LS] Antonin Leroux and Sina Schaeffler. New algorithms for quaternion ideals in sqsign. to appear. 2026.
- [NS04] Phong Q. Nguyen and Damien Stehlé. Low-dimensional lattice basis reduction revisited. In Duncan Buell, editor, *Algorithmic Number Theory*, pages 338–357, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
- [NS09] Phong Q. Nguyen and Damien Stehlé. An lll algorithm with quadratic complexity. *SIAM Journal on Computing*, 39(3):874–903, 2009.
- [PLL19] Christian Porter, Shanxiang Lyu, and Cong Ling. On the optimality of gauss’s algorithm over euclidean imaginary quadratic fields. In *2019 IEEE Information Theory Workshop (ITW)*, pages 1–5, 2019.
- [PR23] Aurel Page and Damien Robert. Introducing clapoti(s): Evaluating the isogeny class group action in polynomial time. Cryptology ePrint Archive, Paper 2023/1766, 2023.
- [SD16] Noah Stephens-Davidowitz. Search-to-Decision Reductions for Lattice Problems with Approximation Factors (Slightly) Greater Than One. In Klaus Jansen, Claire Mathieu, José D. P. Rolim, and Chris Umans, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2016)*, volume 60 of

Leibniz International Proceedings in Informatics (LIPIcs), pages 19:1–19:18, Dagstuhl, Germany, 2016. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

- [Val91] Brigitte Vallée. Gauss’ algorithm revisited. *Journal of Algorithms*, 12(4):556–572, 1991.
- [vdW] B. L. van der Waerden. Die reduktionstheorie der positiven quadratischen formen. 96(1):265–309.