SQIsign with Fixed-Precision Integer Arithmetic

Won Kim, Jeonghwan Lee, Hyeonhak Kim, and Changmin Lee

Korea University

Abstract. SQIsign is an isogeny-based post-quantum signature scheme over supersingular elliptic curves that represents isogenies as elements of a quaternion algebra, enabling highly compact signatures and efficient computation. However, because SQIsign performs quaternion arithmetic over $\mathbb{Q}$, no explicit, uniform worst-case bound is available for the integer coefficients used to represent quaternion algebra elements. Hence, existing implementations require multi-precision integer arithmetic which hinders portability and complicates memory management, disabling constant-time and embedded-friendly implementations.

In this work, we perform a complete analysis of all routines in the Round-2 SQIsign specification that manipulate quaternion elements and establish an explicit uniform worst-case size bound, with a hypothesis to make all intermediate quaternion computations provably bounded. This proof removes the need for multi-precision arithmetic, enabling the first implementation of SQIsign with fixed-precision integer arithmetic, further presenting possibility of constant-time and memory-friendly implementation.

We further tighten this bound by introducing a modified ideal multiplication algorithm, which is a subroutine of SQisign. By modifying the ideal multiplication, we derived the improvement of the size of uniform bound compared with the experimental maximum bit of original Round-2 SQIsign, as 45%/44%/44.5%, for NIST-I/III/V security levels, respectively. Relying on the reduced uniform bound, we build a fixed-precision C implementation of SQIsign.

Keywords: SQIsign · Quaternion Algebra · Worst-Case Bound · Fixed-Precision Integer Arithmetic.

1 Introduction

The National Institute of Standards and Technology (NIST) initiated the Post-Quantum Cryptography (PQC) standardization process in response to the threat that quantum computers pose to widely used public-key systems. First, NIST selected CRYSTALS-Kyber [1] as the standard Key Encapsulation Mechanism (KEM) and CRYSTALS-Dilithium [2], Falcon [3], and SPHINCS+ [4] as digital signature schemes. NIST released the three finalized standards: FIPS 203 (ML-KEM), FIPS 204 (ML-DSA), and FIPS 205 (SLH-DSA). Afterward, NIST selected HQC [5], a code-based KEM for standardization as an additional KEM algorithm. To avoid reliance on a single mathematical hardness assumption for

signatures except for SLH-DSA, an inefficient algorithm, NIST launched an additional call for digital signatures grounded in diverse mathematical hard problems. Currently, 14 schemes, including SQIsign, are candidates in Round-2.

Post-Quantum signatures for constrained platforms demand compact artifacts and predictable memory use. SQIsign [6] achieves the best compactness among Post-Quantum signature schemes, by representing isogenies with quaternion algebra techniques via Deuring's correspondence [7]. The sizes of public keys and signatures of SQIsign are dramatically smaller than them of the NIST's finalized standard signatures (ML-DSA, SLH-DSA) and selected digital signature (Falcon). A summary for their sizes is given in Table 1. This compactness has many advantages: it shortens TLS handshakes [8] and X.509 chains [9], and it means smaller buffers and certificate structures to parse and store, which fits constrained devices and motivates IoT-oriented TLS/X.509 profiles and compact encodings [10]. Moreover, SQIsign is the unique isogeny-based signature advanced to NIST's Round-2. Unlike some SIDH-like schemes broken by the attack using 2D-isogeny [11, 12, 13], SQIsign not only remains secure but also benefits from 2D-isogeny for speedups [14]. Since SQIsign is the fastest isogeny-based signature scheme and its security is proven [15], it is the most attractive isogeny-based digital signature scheme for many researchers.

Table 1: Public key (PK) and signature (Sig) sizes in bytes at NIST security levels I/III/V. Falcon defines parameter sets at levels I and V only.

	PK (bytes)			Sig (bytes)		
Scheme	I	III	V	I	III	V
SQIsign	65	97	129	148	224	292
ML-DSA (Dilithium)	1312	1952	2592	2420	3309	4627
Falcon	897	—	1793	666	—	1280
SLH-DSA (SPHINCS+)	32	48	64	7856	16224	29792

In the NIST PQC process, implementation discipline is a first-order objective. The Call for Additional Digital Signature Schemes explicitly values side-channel aware, constant-time implementations over those that do not address, and NIST status reports emphasize benchmarking on constrained devices with concrete resource costs, not just speed. Consistent with this emphasis, NIST's third-round report presents Cortex-M4 results and the second-round report highlights the value of ARM Cortex-M4 implementations for understanding algorithmic performance and memory requirements.

However, in SQIsign, consisting of KeyGen/Sign/Verify, the implementation by multi-precision complicates constant-time guarantees and portability on memory-restricted embedded devices. It is because multi-precision integer arithmetic such as GMP makes control flow and memory traffic depend on operand size [16], whereas a fixed-precision design enforces secret-independent execution and de-

terministic memory footprints that suit microcontrollers. Since no provable size limit exists for quaternion coefficients, implementers must use dynamic precision to avoid overflow.

Moreover, the original Round-2 SQIsign algorithm does not tightly consider the necessary memory size of integer arithmetic. Even if one derives the uniform worst-case bound of the original Round-2 SQIsign, the result tends to be conservative. There is a subroutine IdealMultiplication of KeyGen and Sign of SQIsign, in which the growth of the size bound is powerful. The original Round-2 SQIsign computes it naively so it is fast but needs too much memory for integer arithmetic.

We summarize the challenges in SQIsign related to memory-friendly implementation to solve.

Challenges

- **No explicit uniform worst-case bounds for the quaternion operations.** The Round-2 specification and prior implementations provide no input-independent bounds on the growth of the integers used to represent quaternion elements along the KeyGen and Sign paths. Computing the uniform bound of quaternion algebra is difficult because it is a structure over $\mathbb{Q}$, which is an infinite algebraic structure. So there is some bottleneck such as the reduced norm of secret key ideal has no proven upper bound. This has forced the use of multi-precision libraries such as GMP in SQIsign.

- **Excessive growth in IdealMultiplication.** In SQIsign, the algorithm IdealMultiplication multiplies two ideals $I_1 = \langle \alpha_1, \alpha_2, \alpha_3, \alpha_4 \rangle$ and $I_2 = \langle \beta_1, \beta_2, \beta_3, \beta_4 \rangle$, where α_s, β_t are all in a quaternion algebra over $\mathbb{Q}$. It is implemented by clearing denominators to common factors r_1, r_2, forming the 4×16 matrix of coefficients of the products $r_1\alpha_s \cdot r_2\beta_t$ in the $\{1, i, j, k\}$ basis, and computing its Hermite normal form (HNF).
  The sizes of intermediates during HNF can be governed by the g.c.d(greatest common divisor) of 4×4 minors of this matrix. Computing all minors needs $\binom{16}{4}$ computations of determinants, which is inefficient. So, in the original Round-2 algorithm, it computes just 4 minors, whose g.c.d is usually larger than the one of $\binom{16}{4}$ minors. It results in HNF performed modulo an oversized determinant modulus, which leads to numerators that swell dramatically and forces much more fixed-precision number of limbs(c.f. words) than the one actually we need.

1.1 Our Contribution

We addressed these challenges with two contributions:

– **Theoretically proven uniform bound for KeyGen and Sign.** We analyze all routines that manipulate quaternion elements, derive per-routine bounds, and compose them into explicit uniform worst-case bounds for the full KeyGen/Sign pipelines. We assume that the reduced norm of the secret key ideal is less than the SQIsign parameter p, to make all intermediate ideals and quaternion elements of SQIsign provably boundable. We additionally assume each input basis of IdealMultiplication is LLL-reduced, to get a bound for the norm of each basis element. If there is no such a condition, there is no size bound for any arbitrary basis elements, it seems to be impossible to get a uniform bound. As a result, we conclude the uniform-bound for KeyGen and Sign as following: (See Section 3 for details)

Proposition 1 (Original KeyGen bound). *In original Round-2 SQIsign, let p be a parameter of SQIsign depending on the security parameter λ. While executing **KeyGen**, suppose that each input basis of ideal multiplication is δ-LLL-reduced for any arbitrary $\frac{3}{4} \leq \delta < 1$. Then, the maximum size of the integers during the operations over a quaternion algebra is less than*

$$2^{50} p^{14} (\log p)^{96}.$$

Proposition 2 (Original Sign bound). *In original Round-2 SQIsign, let p be a parameter of SQIsign depending on the security parameter λ. While executing **Sign**, suppose that each input basis of ideal multiplication is δ-LLL-reduced for any arbitrary $\frac{3}{4} \leq \delta < 1$. Then, the maximum size of the integers during the operations over a quaternion algebra is less than*

$$2^{50} p^{95}.$$

Having such a uniform bound fixes a single, input-independent bitlength budget for all intermediates. In turn, this enables a fixed-precision implementation with data-independent control flow and predictable memory usage.

– **Tighter bound by modification of ideal multiplication.** We redesign IdealMultiplication to reduce the size of the uniform bound. This modification controls numerator growth at each step, in contrast to the original Round-2 SQIsign, and yields substantially tighter closed-form bounds (see Section 4.1). By this modification, we derive a uniform bound, which is improved upon the one of original Round-2 SQIsign, in two respects. First, it can be derived removing the condition that each input basis of IdealMultiplication is LLL-reduced. On such a condition, one should compute LLL-reduced bases before each IdealMultiplication in implementational level. It would result in overhead. Second, the modification of IdealMultiplication reduces the size of the uniform bound. Our uniform bound for KeyGen and Sign with modification of ideal multiplication is as following: (See Section 4.3 for details)

Theorem 1 (Modified KeyGen bound). *In SQIsign with modified **Ideal-Multiplication**, let p be a parameter of SQIsign depending on the security parameter λ. While executing **KeyGen**, the maximum value of the size of integers during the operations over a quaternion algebra is less than*

$$2^{68}p^9.$$

Theorem 2 (Modified Sign bound). *In SQIsign with modified **Ideal-Multiplication**, let p be a parameter of SQIsign depending on the security parameter λ. While executing **Sign**, the maximum value of the size of integers during the operations over a quaternion algebra is less than*

$$2^{16}p^{28}.$$

Since $p \approx 2^{2\lambda}$ where $\lambda = 128/192/256$ for each security level [17], the modified uniform bound is meaningfully smaller than the original uniform bound. We compare our uniform bound after the modification of IdealMultiplication and the experimental bound of the original SQIsign implementation, to assess how our modification of IdealMultiplication reduces the size bound. (Theoretical uniform bound of original Round-2 SQIsign needs a condition about the input basis of IdealMultiplication.) Table 2 shows the comparison of the experimental bound of the original Round-2 SQIsign implementation and the theoretically proven uniform bound of our modified SQIsign. We provide a C implementation demonstrating feasibility, applying our modification of IdealMultiplication. (See Appendix B.)

Table 2: Improvement of size bound after the modification of IdealMultiplication.

		NIST-I	**NIST-III**	**NIST-V**
Before Modification	Experimental maximum bit	12,755	19,194	25,539
	Necessary bit size	12,800	19,200	25,600
After Modification	Uniform maximum bit	7,026	10,713	14,150
	Necessary bit size	7,040	10,752	14,208
	Improvement	45%	44%	44.5%

1.2 Future Work

With explicit, uniform worst-case bounds established for the quaternion layer and a GMP-free fixed-precision implementation in place, the present work opens several avenues for both engineering maturation and further analysis. We detail

these directions next.

(1) Optimization of fixed-precision arithmetic Optimizing our implementation of fixed-precision integer arithmetic, which is dramatically slower than the GMP-based dynamic-precision integer arithmetic.
(2) Embedded KeyGen/Sign. Porting KeyGen and Sign to microcontrollers such as Cortex-M4, using our fixed-precision design. Prior work and code paths emphasize verification on such targets, extending to full signing will test our sizing under tight RAM/flash budgets and constant-time constraints.
(3) Tighter analytic bounds. Pursuing sharper worst-case bounds via refined geometric-of-numbers arguments and structure-aware norm estimates for quaternion ideals, aiming to relax basis assumptions while keeping bounds explicit.
(4) Separate fixed-precision budgets for KeyGen/Sign Separating numbers of limbs of fixed-precision arithmetic for KeyGen/Sign to yield measurable RAM savings on embedded targets, because the uniform bound for KeyGen is much less than Sign, by Theorem 1 and Theorem 2.

2 Preliminaries

In this section, we give some background knowledge about SQIsign and lattices in a quaternion algebra over $\mathbb{Q}$. Note that some lemmas we proved but not essential are described in Appendix A. We also listed up some statements cited from references in Appendix G.

Notations. We denote *"size"* as the biggest integer appearing in elements of quaternion algorithms for SQIsign. (e.g. denominators and numerators of coefficietns of quaternion elements) We also use parameters defined in [17]. (Also, see Appendix E.) For example, $p \equiv 3 \pmod 4$ is a prime with $p \approx 2^{2\lambda}$ where λ is a security parameter.

2.1 Quaternion Algebras

We denote $B_{p,\infty} = \left(\frac{-1,-p}{\mathbb{Q}}\right)$, the quaternion algebra over $\mathbb{Q}$ ramified exactly at p and ∞. It is a 4-dimensional $\mathbb{Q}$-vector space with basis $\{1, i, j, k := ij\}$, $i^2 = -1,\ \ j^2 = -p,\ \ ij = -ji$. In the rest of this paper, we denote p as a prime such that $p \equiv 3 \pmod 4$.

For $\alpha = a + bi + cj + dk \in B_{p,\infty}$, define the conjugate $\bar{\alpha} := a - bi - cj - dk$, and the reduced trace and norm $\mathrm{trd}(\alpha) := \alpha + \bar{\alpha} = 2a$, $\mathrm{nrd}(\alpha) := \alpha\bar{\alpha} = a^2 + b^2 + pc^2 + pd^2$. An **order** $\mathcal{O} \subset B_{p,\infty}$ is a subring that is a finitely generated $\mathbb{Z}$-module spanning $B_{p,\infty}$ over $\mathbb{Q}$. It is *maximal* if it is not properly contained in any larger order of $B_{p,\infty}$.

2.2 Lattices and Ideals

Each left ideal $I \subset B_{p,\infty}$ forms a full-rank lattice in $\mathbb{R}^4$ under the standard embedding, with norm and trace given by the reduced norm and trace of the quaternion algebra.

Let $B = (b_1, b_2, \cdots, b_n)$ be a basis and $G = B^T B$ its gram matrix. For lattices we use the **Hermite Normal Form (HNF)**, analogous to the reduced echelon form for matrices. Let $A \in \mathbb{Z}^{m\times n}$ be an integer matrix of full row rank m. A matrix $U \in \mathbb{Z}^{m\times m}$ is called *unimodular* if $\det(U) = \pm 1$ (equivalently, $U \in \mathrm{GL}_m(\mathbb{Z})$).

Definition 1. *An integer matrix $H \in \mathbb{Z}^{m\times n}$ is said to be in* Hermite Normal Form (HNF) *if:*

1. *H is upper triangular, and any rows of zeros are located below any other row.*
2. *The first nonzero entry from the left(=pivot) of a nonzero row is positive and always strictly right to the pivot of the row above it.*
3. *The elements to the left of pivots are zero and elements to the right of pivots are nonnegative and smaller than the pivot.*

We also use **LLL reduction** to obtain an LLL-reduced basis, which is short, nearly orthogonal basis. When $(b_1, b_2, \cdots, b_n)$ is a δ-LLL-reduced basis of L for some $\frac{3}{4} \leq \delta < 1$, there is a property that

$$\prod_{i=1}^{n} \|b_i\| \leq 2^{n(n-1)/4} \det(L).$$

For an order $\mathcal{O}$ in a quaternion algebra, a **left (fractional) $\mathcal{O}$-ideal** I is a finitely generated $\mathbb{Z}$-module stable under left multiplication by $\mathcal{O}$—hence, I forms a lattice [18]. For any full-rank lattice $I \subset B_{p,\infty}$, we define:

$$O_L(I) := \{\alpha \in B_{p,\infty} \mid \alpha I \subseteq I\}, \quad O_R(I) := \{\alpha \in B_{p,\infty} \mid I\alpha \subseteq I\},$$

as the *left* and *right orders* of I, respectively. Both are maximal orders in $B_{p,\infty}$. This property ensures that ideal multiplication I_1, I_2 is well-defined when $O_R(I_1) = O_L(I_2)$. The (reduced) norm of I is given by

$$\mathrm{nrd}(I) := \gcd\{\, \mathrm{nrd}(\alpha) \mid \alpha \in I \,\}.$$

If I is also a right $\mathcal{O}'$-ideal, it is called an $(\mathcal{O}, \mathcal{O}')$-*connecting ideal*. Two left $\mathcal{O}$-ideals I, J are said to be **equivalent** if there exists $\alpha \in B^\times$ such that $J = I\alpha$. Moreover, for any nonzero $\alpha \in I$, the ideal $I \cdot (\bar{\alpha}/\mathrm{nrd}(\alpha))$ is equivalent to I [18].

Remark 1. In particular, an isomorphism between ideals sends a reduced basis to a reduced basis. The normalized norm map

$$q_I := \frac{\mathrm{nrd}(\cdot)}{\mathrm{nrd}(I)} : I \longrightarrow \mathbb{Z}$$

is invariant under this isomorphism, in the sense that $q_I(\alpha) = q_J(\beta)$ whenever $\beta = \alpha\gamma$ for some $\gamma \in B^\times$.

2.3 Elliptic Curves, Isogenies and Deuring Correspondence

An **elliptic curve** E/K is a smooth projective genus-1 curve with a distinguished K-rational point O. An **isogeny** $\varphi : E \to E'$ is a non-constant morphism that is also a group homomorphism of $\bar{K}$-rational points. It satisfies $\deg(\varphi \circ \psi) = \deg(\varphi)\deg(\psi)$ and admits a unique dual $\hat{\varphi} : E' \to E$ with $\hat{\varphi} \circ \varphi = [\deg \varphi]_E$ and $\varphi \circ \hat{\varphi} = [\deg \varphi]_{E'}$. For an elliptic curve E/K, we write a ring

$$\mathrm{End}(E) := \{\varphi : E \to E \mid \varphi \text{ is a group homomorphism over } \bar{K}\}$$

for the **endomorphism ring** of E, with pointwise addition and composition as multiplication. An elliptic curve $E/\bar{\mathbb{F}}_p$ is **supersingular** if $\mathrm{End}(E)$ is isomorphic to a maximal order in $B_{p,\infty}$. A canonical example is $E_0 : y^2 = x^3 + x$ over $\mathbb{F}_{p^2}$, for $p \equiv 3 \pmod 4$, whose endomorphism ring corresponds to a maximal order $\mathcal{O}_0 \subset B_{p,\infty}$.

Deuring Correspondence. For primes $p \equiv 3 \pmod 4$, there exists a bijection between isomorphism classes of supersingular elliptic curves over $\bar{\mathbb{F}}_p$ (up to $\mathbb{F}_{p^2}$-isomorphism) and maximal orders in $B_{p,\infty}$ [7]. Moreover, morphisms and structural operations translate as follows:

Elliptic-curve world	**Quaternion-algebra world**
E (supersingular)	$\mathcal{O} = \mathrm{End}(E) \subset B_{p,\infty}$
$\varphi : E \to E'$	Left $\mathcal{O}$-ideal and right $\mathcal{O}'$-ideal I_φ
$\deg(\varphi)$	$\mathrm{nrd}(I_\varphi)$
$\varphi, \psi : E \to E'$ equivalent	$I_\varphi \sim I_\psi$
$\hat{\varphi} : E' \to E$	$I_{\hat{\varphi}}$
$\varphi \circ \psi$	$I_\psi \cdot I_\varphi$

This correspondence forms the algebraic foundation for representing isogenies in terms of quaternion ideals, a central idea underlying the design of SQIsign.

2.4 SQIsign

SQIsign is a post-quantum digital signature whose security is tied to the difficulty of computing an isogeny path $\varphi : E \to E'$ given domain and image supersingular elliptic curves E, E' over $\mathbb{F}_{p^2}$ [15, 19, 20]. At a high level, SQIsign builds a Σ-protocol for the proof of knowledge of an isogeny and applies Fiat-Shamir transform. Its concrete instantiation relies on quaternion maximal orders in $B_{p,\infty}$, ideals, and explicit isogeny arithmetic together with lattice subroutines.

The syntax of the digital signature scheme SQIsign is defined as follows:
$\mathsf{Setup}(1^\lambda) \to \mathsf{params}$: Given a security parameter λ, outputs public parameters params.
$\mathsf{KeyGen}(\mathsf{params}) \to (\mathsf{pk}, \mathsf{sk})$: Generates a secret ideal I_{sk} in the quaternion algebra $B_{p,\infty}$ and derives the corresponding public key through the isogeny φ_{sk}.

$\mathsf{Sign}(\mathsf{sk}, m) \to \sigma$: Produces a signature σ as a non-interactive proof of knowledge φ_{sk} using the Fiat-Shamir transform.
$\mathsf{Verify}(\mathsf{pk}, m, \sigma) \to \{\mathsf{accept}, \mathsf{reject}\}$: Recomputes the challenge hash and verifies σ against (pk, m).

We also give a diagram for KeyGen and Sign of SQIsign. It represents dependencies between ideal multiplication, subroutines of KeyGen and Sign, KeyGen and Sign of SQIsign. We use a dotted line to indicate a conditional dependency, (since KeyGen does not compute ideal multiplication directly) and arrows to represent each direction from a required subroutine to an algorithm which uses that subroutine. (Fig 1)

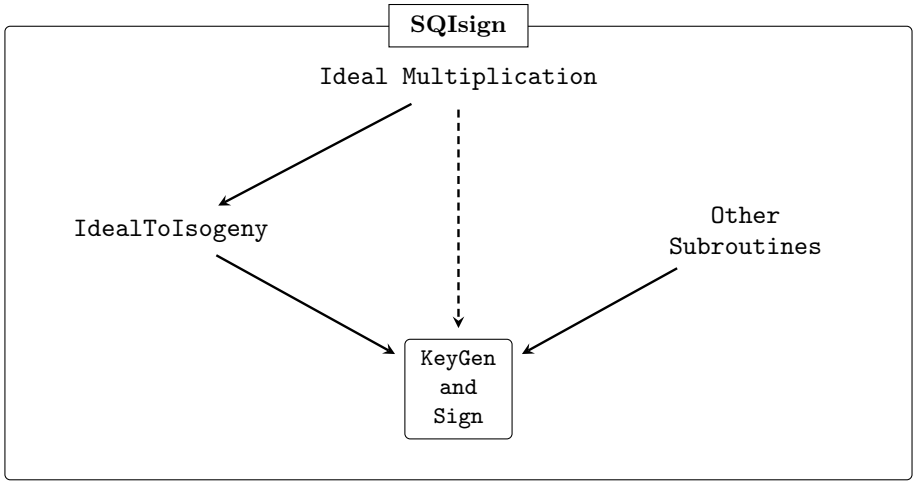

Fig. 1: Diagram of KeyGen/Sign of SQIsign.

3 The Uniform Bound of Original Round-2 SQIsign

In this section, we derive uniform worst-case bounds for the integer sizes that arise in the original Round-2 SQIsign. We first establish individual bound of subroutines **IdealMultiplication**, **SuitableIdeals IdealToIsogeny**, and **RandomIdealGivenNorm** that performs arithmetic over a quaternion algebra. We give the bounds for other subroutines in Appendix C. Then we integrate these per-routine results to obtain the uniform bound for the overall **KeyGen** and **Sign** of original Round-2 SQIsign.

3.1 IdealMultiplication

We begin by analyzing the maximum size growth caused by ideal multiplication since it governs the eventual worst-case bounds for both **KeyGen** and **Sign** of

Algorithm 1 HNF(M)

Input: An integer matrix M with d rows and $c \geq d$ columns with rank d. Its columns are denoted by M_1^{t} to M_c^{t}, and the coefficient in row r and column l is denoted by $M_{r,l}$.
Output: The HNF of M.

for i **from** d **down to** 1 **do**
  for j **from** $i-1$ **down to** 1 **do**
    if $M_{i,i}$ and $M_{i,j}$ are both 0 **then**
      $g, u, v \leftarrow 1, 1, 0$
    else
      $g, u, v \leftarrow \mathrm{XGCD}(M_{i,i}, M_{i,j})$ ▷ After this step, $uM_{i,i} + vM_{i,j} = g$ and $u > 0$
    end if
    $M_i^{\mathrm{t}} \leftarrow uM_i^{\mathrm{t}} + vM_j^{\mathrm{t}}$ ▷ After this step $M_{i,i}$ equals g
  end for
  for j **from** $i-1$ **down to** 1 **do**
    $g \leftarrow M_{i,j}/M_{i,i}$ ▷ g is an integer
    $M_j^{\mathrm{t}} \leftarrow M_j^{\mathrm{t}} - gM_i^{\mathrm{t}}$ ▷ After this step $M_{i,j} = 0$
  end for
  for j **from** $i+1$ **up to** c **do**
    $r \leftarrow M_{i,j} \bmod M_{i,i}$ ▷ After this $r \in [0, M_{i,i} - 1]$
    $g \leftarrow (M_{i,j} - r)/M_{i,i}$ ▷ g is an integer
    $M_j^{\mathrm{t}} \leftarrow M_j^{\mathrm{t}} - gM_i^{\mathrm{t}}$ ▷ After this step $M_{i,j} = r \in [0, M_{i,i} - 1]$
  end for
end for
return M

SQIsign. **IdealMultiplication** is an algorithm computing the multiplication of two ideals given their bases.

In SQIsign, computing the multiplication of two ideals I_1, I_2 is performed by computing **HNF** of 16 generators of the result ideal [21]. The algorithm **HNF** for computing the HNF of a given integer matrix, is as follows: (Algorithm 1)

In more detail, when we compute $I_1 I_2$ for $I_1 = \langle \alpha_1, \alpha_2, \alpha_3, \alpha_4 \rangle$ and $I_2 = \langle \beta_1, \beta_2, \beta_3, \beta_4 \rangle$, first we multiply I_1, I_2 by $\mathrm{lcm}(r_{\alpha_1}, r_{\alpha_2}, r_{\alpha_3}, r_{\alpha_4})$, $\mathrm{lcm}(r_{\beta_1}, r_{\beta_2}, r_{\beta_3}, r_{\beta_4})$, respectively, to make each generator can be represented in $\mathbb{Z}^4$. Then we have each generator which is a quaternion element with integer coefficients so we can represent it as $\gamma \mapsto (a_\gamma, b_\gamma, c_\gamma, d_\gamma)$. Concretely, we compute **HNF** of a 4×16 matrix M whose entries $M_{1,l}, M_{2,l}, M_{3,l}, M_{4,l}$ are $a_{\alpha_{\lceil \frac{l}{4} \rceil} \beta_{(l \bmod 4)+1}}$, $b_{\alpha_{\lceil \frac{l}{4} \rceil} \beta_{(l \bmod 4)+1}}$, $c_{\alpha_{\lceil \frac{l}{4} \rceil} \beta_{(l \bmod 4)+1}}$, $d_{\alpha_{\lceil \frac{l}{4} \rceil} \beta_{(l \bmod 4)+1}}$, respectively.

To derive the bound for **IdealMultiplication**, we establish the bound for the input of **HNF** first.

Lemma 1. *Let* $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}, \{\beta_1, \beta_2, \beta_3, \beta_4\}$ *be bases of ideals* I_1, I_2 *in* $B_{p,\infty}$ *with integer coefficients, respectively, where* $O_R(I_1) = O_L(I_2)$*. Let* M *be a* 4×16 *integer matrix whose each entry* $M_{r,l}$ *is*

$$M_{1,l} = a_{\alpha_{\lceil \frac{l}{4} \rceil} \beta_{(l-1) \bmod 4+1}}, M_{2,l} = b_{\alpha_{\lceil \frac{l}{4} \rceil} \beta_{(l-1) \bmod 4+1}},$$
$$M_{3,l} = c_{\alpha_{\lceil \frac{l}{4} \rceil} \beta_{(l-1) \bmod 4+1}}, M_{4,l} = d_{\alpha_{\lceil \frac{l}{4} \rceil} \beta_{(l-1) \bmod 4+1}}.$$

for each integer $1 \leq l \leq 16$*. Then,*

$$|M_{1,l}|, |M_{2,l}| \leq \sqrt{\operatorname{nrd}\left(\alpha_{\lceil \frac{l}{4} \rceil}\right) \operatorname{nrd}(\beta_{(l-1) \bmod 4+1})},$$
$$|M_{3,l}|, |M_{4,l}| \leq \frac{1}{\sqrt{p}} \sqrt{\operatorname{nrd}\left(\alpha_{\lceil \frac{l}{4} \rceil}\right) \operatorname{nrd}(\beta_{(l-1) \bmod 4+1})}.$$

Proof. See Appendix A.1. □

Before multiplying two ideals, since the basis elements may have rational coefficients, we first clear denominators so that the corresponding basis matrix has integral entries and HNF can be applied. To compute **HNF** to multipliy two ideals. Then, we apply **HNF** to the matrix $r_1 r_2 M$, where

$$r_1 = \operatorname{lcm}(r_{\alpha_1}, r_{\alpha_2}, r_{\alpha_3}, r_{\alpha_4}),$$
$$r_2 = \operatorname{lcm}(r_{\beta_1}, r_{\beta_2}, r_{\beta_3}, r_{\beta_4}).$$

By this substitution, when we apply the result of Lemma 1, we just need to apply Lemma 1 by letting values $\operatorname{nrd}(\alpha_s)\operatorname{nrd}(\beta_t)$ be $r_1^2 r_2^2 \operatorname{nrd}(\alpha_s)\operatorname{nrd}(\beta_t)$.

During the ideal multiplication, we need to prevent the entries of the intermediate matrices in the HNF computation from growing exponentially (see Appendix F). To control the size of the coefficients, we perform all arithmetic of the **HNF** algorithm in the quotient ring $\mathbb{Z}_m := \mathbb{Z}/m\mathbb{Z}$, that is, after each operation every matrix entry is reduced modulo m and hence has absolute value strictly less than m. Such a modular version of the HNF algorithm is available over $\mathbb{Z}_m$ [22, 23], where we may take

$$m \;=\; t \cdot \gcd\{\det(S) \mid S \text{ is a } 4 \times 4 \text{ submatrix of } M\}$$

for some $t \in \mathbb{Z}_{>0}$.

In original Round-2 SQIsign, multiplication of two ideals is performed as follows: (Algorithm 2)

In **line 3** of Algorithm 2, we can compute a determinant of 4×4 matrix with determinants of 2×2 submatrices [24]. (See Algorithm 3)

Bound on IdealMultiplication. We derive the size bound for numerators in **IdealMultiplication** as Lemma 2, and the size bound for denominators as Lemma 5. We assume that the bases of I_1 and I_2 are δ-LLL-reduced (with

Algorithm 2 IdealMultiplication(I_1, I_2) (Original Round-2 SQIsign Version)

Input: Maximal orders $\mathcal{O}_1, \mathcal{O}_2$ in $B_{p,\infty}$, a left $\mathcal{O}_1$-ideal $I_1 = \langle \alpha_1, \alpha_2, \alpha_3, \alpha_4 \rangle$ and a left $\mathcal{O}_2$-ideal $I_2 = \langle \beta_1, \beta_2, \beta_3, \beta_4 \rangle$ such that $O_R(I_1) = O_L(I_2)$.
Output: A left $\mathcal{O}_1$-ideal $I_1 I_2 = \langle \gamma_1, \gamma_2, \gamma_3, \gamma_4 \rangle$.
1: $r_1 \leftarrow \mathrm{lcm}(r_{\alpha_1}, r_{\alpha_2}, r_{\alpha_3}, r_{\alpha_4})$, $r_2 \leftarrow \mathrm{lcm}(r_{\beta_1}, r_{\beta_2}, r_{\beta_3}, r_{\beta_4})$
2: $\alpha_i \leftarrow r_1 \alpha_i$, $\beta_j \leftarrow r_2 \beta_j$ for $1 \le i, j \le 4$
3: $m \leftarrow |\det(\alpha_1\beta_1\ \alpha_1\beta_2\ \alpha_1\beta_3\ \alpha_1\beta_4)|$
4: Compute $M \leftarrow \mathbf{HNF}(\alpha_i \beta_j)_{1 \le i,j \le 4}$ over $\mathbb{Z}_m$.
5: **return** $\frac{1}{r_1 r_2} M$

Algorithm 3 Determinant(M)

Input: A 4×4 integer matrix M.
Output: $\det(M)$.
1: $s_0 \leftarrow \det \begin{pmatrix} M_{0,0} & M_{0,1} \\ M_{1,0} & M_{1,1} \end{pmatrix}$, $s_1 \leftarrow \det \begin{pmatrix} M_{0,0} & M_{0,2} \\ M_{1,0} & M_{1,2} \end{pmatrix}$, $s_2 \leftarrow \det \begin{pmatrix} M_{0,0} & M_{0,3} \\ M_{1,0} & M_{1,3} \end{pmatrix}$
$s_3 \leftarrow \det \begin{pmatrix} M_{0,1} & M_{0,2} \\ M_{1,1} & M_{1,2} \end{pmatrix}$, $s_4 \leftarrow \det \begin{pmatrix} M_{0,1} & M_{0,3} \\ M_{1,1} & M_{1,3} \end{pmatrix}$, $s_5 \leftarrow \det \begin{pmatrix} M_{0,2} & M_{0,3} \\ M_{1,2} & M_{1,3} \end{pmatrix}$
2: $c_0 \leftarrow \det \begin{pmatrix} M_{2,0} & M_{2,1} \\ M_{3,0} & M_{3,1} \end{pmatrix}$, $c_1 \leftarrow \det \begin{pmatrix} M_{2,0} & M_{2,2} \\ M_{3,0} & M_{3,2} \end{pmatrix}$, $c_2 \leftarrow \det \begin{pmatrix} M_{2,0} & M_{2,3} \\ M_{3,0} & M_{3,3} \end{pmatrix}$
$c_3 \leftarrow \det \begin{pmatrix} M_{2,1} & M_{2,2} \\ M_{3,1} & M_{3,2} \end{pmatrix}$, $c_4 \leftarrow \det \begin{pmatrix} M_{2,1} & M_{2,3} \\ M_{3,1} & M_{3,3} \end{pmatrix}$, $c_5 \leftarrow \det \begin{pmatrix} M_{2,2} & M_{2,3} \\ M_{3,2} & M_{3,3} \end{pmatrix}$
3: **return** $s_0 c_5 - s_1 c_4 + s_2 c_3 + s_3 c_2 - s_4 c_1 + s_5 c_0$

$\frac{3}{4} \le \delta < 1$) to control the determinant in **line 3** of Algorithm 2. LLL-reduction yields an upper bound on this determinant, and hence on m, in terms of the reduced norms of the basis elements.

Lemma 2. *Let $I_1 = \langle \alpha_1, \alpha_2, \alpha_3, \alpha_4 \rangle$ and $I_2 = \langle \beta_1, \beta_2, \beta_3, \beta_4 \rangle$ be left ideals such that $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ and $\{\beta_1, \beta_2, \beta_3, \beta_4\}$ are δ-LLL-reduced bases with $\frac{3}{4} \le \delta < 1$. During the ideal multiplication routine of original Round-2 SQIsign, the maximum size of the integers that appear is bounded by*

$$2^{10}\, p^{10}\, r_1^{32}\, r_2^{8}\, \mathrm{nrd}(I_1)^{16}\, \mathrm{nrd}(I_2)^{4},$$

where $r_1 := \mathrm{lcm}(r_{\alpha_1}, \ldots, r_{\alpha_4})$ and $r_2 := \mathrm{lcm}(r_{\beta_1}, \ldots, r_{\beta_4})$.

Proof. Let r_1, r_2 be as in the statement and replace each basis element by $r_1\alpha_i$, $r_2\beta_j$ so that all coordinates are integral. Form the 4×16 matrix M whose ℓ-th column is the coordinate vector of $\alpha_{\lceil \ell/4 \rceil} \beta_{(\ell-1) \bmod 4+1}$ in the basis $(1, i, j, k)$; equivalently, we apply HNF to $r_1 r_2 M$ over $\mathbb{Z}/m\mathbb{Z}$ for a suitable modulus m.

Step 1. (bound on the determinant modulus). By Hadamard's inequality,

$$m \;=\; \big|\det([\alpha_1\beta_1\ \alpha_1\beta_2\ \alpha_1\beta_3\ \alpha_1\beta_4])\big| \;\le\; \prod_{i=1}^{4} \|\alpha_1\beta_i\| \;\le\; \Big(\prod_{i=1}^{4} \|\alpha_1\|\Big)\Big(\prod_{i=1}^{4} \|\beta_i\|\Big).$$

Since both input bases are δ-LLL-reduced with $\frac{3}{4} \le \delta < 1$, we have $\prod_{i=1}^{4} \|\alpha_i\| \le (8 \det(r_1 I_1))^4$ and $\prod_{i=1}^{4} \|\beta_i\| \le 8 \det(r_2 I_2)$ so

$$m \;\le\; \big(8 \det(r_1 I_1)\big)^4 \big(8 \det(r_2 I_2)\big).$$

Using $\det(rI) = \det(\mathcal{O}) \operatorname{nrd}(rI)^2$ by Lemma 15 (see Appendix A) and $\det(\mathcal{O}) = \frac{p}{4}$ by Lemma 3, together with $\operatorname{nrd}(rI) = r^2 \operatorname{nrd}(I)$, we obtain

$$m \;\le\; 2^{15} \Big(\frac{p}{4}\Big)^5 r_1^{16} r_2^4 \operatorname{nrd}(I_1)^8 \operatorname{nrd}(I_2)^2 \;\le\; 2^5 p^5 r_1^{16} r_2^4 \operatorname{nrd}(I_1)^8 \operatorname{nrd}(I_2)^2.$$

Step 2. (intermediate size in the 4×4 determinant). When computing the 4×4 determinant in **line 3** via 2×2 minors, the largest intermediate integer is bounded by

$$\begin{aligned} \frac{24}{p} \cdot \prod_{i=1}^{4} \sqrt{\operatorname{nrd}(\alpha_1) \operatorname{nrd}(\beta_i)} \;&=\; \frac{24}{p} \cdot \Big(\prod_{i=1}^{4} \|\alpha_1\|\Big)\Big(\prod_{i=1}^{4} \|\beta_i\|\Big) \\ &\le\; 2^8 \cdot 3\, p^4 r_1^{16} r_2^4 \operatorname{nrd}(I_1)^8 \operatorname{nrd}(I_2)^2, \end{aligned}$$

where we used Lemma 1 to bound matrix entries by the reduced norms of factors and the estimate from Step 1 for the products of norms.

Step 3. (HNF modulo m). In **line 4**, the HNF routine is carried out modulo m, with addition, subtraction, multiplication, and division, so all sizes of intermediate entries remain bounded by m^2 [25]. Therefore, across the entire ideal-multiplication routine the maximum integer size is less than m^2, which, by the bound on m above, gives

$$m^2 \;\le\; \big(2^5 p^5 r_1^{16} r_2^4 \operatorname{nrd}(I_1)^8 \operatorname{nrd}(I_2)^2\big)^2 \;=\; 2^{10} p^{10} r_1^{32} r_2^8 \operatorname{nrd}(I_1)^{16} \operatorname{nrd}(I_2)^4.$$

This is the claimed bound.

□

Lemma 3. *Let $\mathcal{O}$ be a maximal order in $B_{p,\infty}$. Then,*

$$\det(\mathcal{O}) = \frac{p}{4}.$$

Proof. See Appendix A.2. □

Remark 2. Since Lemma 2 needs a condition that input bases are δ-LLL-reduced for some $\frac{3}{4} \le \delta < 1$, we assume that each input basis of **IdealMultiplication** is LLL-reduced, in the rest of Section 3. One can easily observe that given an input basis which is not LLL-reduced, the bound of **IdealMultiplication** can be larger than the one given in Lemma 2. But this approach is reasonable because there are two solutions:

- **trivial solution:** applying LLL algorithm before Idealmultiplication
- **Section 4:** modifying Idealmultiplication by replacing the modulus used for HNF algorithm

Bound on denominators. Let I be a left $\mathcal{O}$-ideal, where $\mathcal{O} = O_R(I_0)$ for a left $\mathcal{O}_0$-ideal I_0. Then, we can bound a denominator of each element of $\mathcal{O}$, by the following two lemmas. Consequently, we can also bound a denominator of each basis element of I which is also an element of $\mathcal{O}$.

Lemma 4. *Let I be a left $\mathcal{O}_0$-ideal such that* $\mathrm{nrd}(I) = N$. *Then,*

$$O_R(I) \subset \frac{1}{N}\mathcal{O}_0.$$

Proof. By Proposition 16.6.15 of [18] (see Appendix G),

$$\bar{I}I = \mathrm{nrd}(I)O_R(I).$$

Since $I \subset \mathcal{O}_0$ and $\bar{I} \subset \mathcal{O}_0$, $\bar{I}I \subset \mathcal{O}_0$. So we have

$$O_R(I) \subset \frac{1}{N}\mathcal{O}_0.$$

□

From Lemma 4, we derive the bound of the denominators of elements in $B_{p,\infty}$.

Lemma 5. *Let $\mathcal{O}$ be a maximal order in $B_{p,\infty}$ with a $\mathcal{O}_0, \mathcal{O}$-connecting ideal I s.t.* $\mathrm{nrd}(I) = N$. *Then, for any $\alpha \in \mathcal{O}$, $r_\alpha \leq 2N$.*

Proof. Let $\alpha \in \mathcal{O}$. Then, by Lemma 4,

$$\alpha = \frac{1}{N}\alpha_0 \text{ for some } \alpha_0 \in \mathcal{O}_0.$$

Since $\mathcal{O}_0 = \langle 1, i, \frac{1+k}{2}, \frac{i+j}{2} \rangle$, it is clear that $r_{\alpha_0} \leq 2$.
Hence, we have

$$r_\alpha \leq 2N.$$

□

3.2 IdealToIsogeny and RandomIdealGivenNorm

We apply the size bound of **IdealMultiplication** proven in Section 3.1 to derive size bounds of the two main subroutines, **IdealToIsogeny** and **RandomIdealGivenNorm** used in **KeyGen** and **Sign** processes of SQIsign. (Size bounds for additional subroutines that do not mainly affect the bounds of **KeyGen** and **Sign** are provided in Appendix C.) These results enable us to derive the uniform bound for **KeyGen** and **Sign** of SQIsign, in Section 3.3.

Algorithm 4 SuitableIdeals(I).

Input: A left $\mathcal{O}_0$-ideal I, a bound $d = \Theta(\sqrt{p})$.
Output: $\beta_1, \beta_2 \in I$ and $u, v \in \mathbb{N}^*$ and $f \leq e$ such that $\gcd(uq_I(\beta_1), vq_I(\beta_2)) = 1$ and $uq_I(\beta_1) + vq_I(\beta_2) = 2^f$.

1: Compute a Minkowski reduced basis $(\alpha_1, \ldots, \alpha_4)$ of I.
2: Let $B_j \leftarrow \left\lfloor \frac{1}{4}\sqrt{\frac{d}{q_I(\alpha_j)}} \right\rfloor$ for $j \in [1; 4]$.
3: Sample $x_j \in [-B_j; B_j]^4$ for $j \in [1; 4]$ and initialize $L \leftarrow [(x_1, \ldots, x_4)]$.
4: **for** $(y_1, \ldots, y_4) \in [-B_1; B_1] \times \cdots \times [-B_4; B_4]$ **do**
5: **for** $(x_1, \ldots, x_4) \in L$ **do**
6: $\beta_1 \leftarrow \sum_{j=1}^4 x_j\alpha_j$ and $\beta_2 \leftarrow \sum_{j=1}^4 y_j\alpha_j$.
7: $d_1 \leftarrow q_I(\beta_1),\ d_2 \leftarrow q_I(\beta_2)$.
8: **if** $d_1 \equiv 1 \pmod 2$ and $d_2 \equiv 1 \pmod 2$ and $\gcd(d_1, d_2) = 1$ **then**
9: $u \leftarrow 2^e d_1^{-1} \bmod d_2$.
10: $v \leftarrow \frac{2^e - ud_1}{d_2}$.
11: **if** $v > 0$ **then**
12: $e' \leftarrow \min(v_2(u), v_2(v))$, $u \leftarrow u/2^{e'}$, $v \leftarrow v/2^{e'}$, and $f \leftarrow e - e'$.
13: **return** $\beta_1, \beta_2, u, v, f$.
14: **else**
15: Append $(y_1, \ldots, y_4)$ to L.
16: **end if**
17: **end if**
18: **end for**
19: **end for**

Bound on SuitableIdeals and IdealToIsogeny. To derive the bound for **IdealToIsogeny**, first we establish the bound for **SuitableIdeals**.

The algorithm **SuitableIdeals**, a subroutine of **IdealToIsogeny**, is as follows: (Algorithm 4)

Lemma 6. *In original Round-2 SQIsign, while executing* ***SuitableIdeals*** *(Algorithm 4), the maximum size of all intermediate integers occurring in the quaternion arithmetic is less than*

$$2^{50}\, p^{10}\, (\log p)^{96}\, \mathrm{nrd}(I)^4.$$

Proof. We follow Algorithm 4 line-by-line and show that the largest growth occurs in the preliminary Minkowski reduction step since all subsequent steps stay beneath the stated bound.

(Line 1 and precomputation). The routine forms a basis of $\overline{J_t}\, I$ (here J_t is the SQIsign parameter ideal) and its Gram matrix, then computes an L2/Minkowski-reduced basis [17]. Each basis element of J_t has reduced norm less than $(\log p)^2$ (See Appendix E). Combining Lemma 2 with Lemma 5, the size of all integers

created while multiplying by $\overline{J_t}$, forming Gram matrices, and running **L2** (see Lemma 21 in Appendix C) is bounded by

$$2^{50}\, p^{10}\, (\log p)^{96}\, \mathrm{nrd}(I)^4.$$

The rest of the proof show that this bound dominates the whole procedure and gives the target bound for **SuitableIdeals**.

(Lines 1–2: sizes of a Minkowski-reduced basis and of B_j). Let $(\alpha_1, \ldots, \alpha_4)$ be the Minkowski-reduced basis returned in Line 1. In dimension 4, a Minkowski-reduced basis attains the successive minima, i.e. $\|\alpha_\ell\| = \lambda_\ell$ for $\ell = 1, \ldots, 4$. By Minkowski's second theorem and the argument of Lemma 48 in [26], we have

$$\lambda_l \;\le\; \frac{8p}{\pi^2}\mathrm{nrd}(I).$$

Hence, For any integer $1 \le l \le 4$, the coefficients satisfy

$$r_{\alpha_l} \le 2, \qquad |a_{\alpha_l}|, |b_{\alpha_l}|, |c_{\alpha_l}|, |d_{\alpha_l}| \;\le\; \frac{\sqrt{8p}}{\pi}\,\sqrt{\mathrm{nrd}(I)}.$$

With $q_I(\alpha) = \mathrm{nrd}(\alpha)/\mathrm{nrd}(I)$, we get

$$q_I(\alpha_j) \le \prod_{j=1}^{4} q_I(\alpha_j) \le \frac{64}{\pi^4} p^2$$

and therefore

$$B_j = \left\lfloor 1/4\sqrt{d/q_I(\alpha_j)} \right\rfloor \;\le\; \frac{2}{\sqrt{d}\,\pi^2}\, p.$$

(Lines 3–6: sampling and forming β_1, β_2). For $|x_j|, |y_j| \le B_j$, we form $\beta_1 = \sum_{j=1}^4 x_j\alpha_j,\ \beta_2 = \sum_{j=1}^4 y_j\alpha_j$. Using the coefficient bounds for the α_j and $|x_j|, |y_j| \le 2p/(\sqrt{d}\pi^2)$, each coefficient of β_1, β_2 obeys

$$|a_{\beta_l}|, |b_{\beta_l}|, |c_{\beta_l}|, |d_{\beta_l}| \;\le\; 2 \cdot \frac{2}{\sqrt{d}\pi^2} p \cdot \frac{16}{\pi^2} p \;=\; \frac{64}{\sqrt{d}\,\pi^4}\, p^2. \qquad (l = 1, 2)$$

These are clearly smaller than the bound already established in the first step.

(Lines 7–10: computing $d_1 = q_I(\beta_1)$, $d_2 = q_I(\beta_2)$ and u, v). By construction d_1, d_2 are odd and $\gcd(d_1, d_2) = 1$. We choose $u \equiv 2^e d_1^{-1} \pmod{d_2}$ with $0 \le u < d_2$ and set $v = (2^e - ud_1)/d_2 \in (0, 2^e)$. Thus

$$u < d_2 \le 2^e, \qquad 0 < v < 2^e,$$

and the modular inverse computation may involve an intermediate quantity less than $d_2^2 \le 2^{2e}$. Hence the integers created in Lines 9–10 are less than 2^{2e}, which is again below the bound from the first step.

Conclusion. All remaining lines perform arithmetic with quantities bounded as above and therefore do not exceed the worst-case size reached in the first (Minkowski reduction) step trivially. This completes the proof.

□

Algorithm 5 IdealToIsogeny(I)

Input: A left $\mathcal{O}_0$-ideal I.
Output: The codomain E_I of φ_I, $\varphi_I(P_0)$, and $\varphi_I(Q_0)$.

1: $u, v, e, s, t, \beta_1, \beta_2 \leftarrow$ **SuitableIdeals**(I)
2: $d_1 \leftarrow \mathrm{nrd}(\beta_1)/\mathrm{nrd}(\overline{J_s}I)$
3: $d_2 \leftarrow \mathrm{nrd}(\beta_2)/\mathrm{nrd}(\overline{J_t}I)$ // $ud_1 + vd_2 = 2^e$, with $\gcd(ud_1, vd_2) = 1$
4: $E_u, \varphi_u(P_s), \varphi_u(Q_s) \leftarrow$ **FixedDegreeIsogeny**(s, u) // $\varphi_u : E_s \to E_u$ is an isogeny of degree u
5: $E_v, \varphi_v(P_t), \varphi_v(Q_t) \leftarrow$ **FixedDegreeIsogeny**(t, v) // $\varphi_v : E_t \to E_v$ is an isogeny of degree v
6: $[P, Q]^T \leftarrow \left[\frac{1}{\mathrm{nrd}(I)\mathrm{nrd}(J_t)}\right] \mathbf{M}_{\beta_1}\mathbf{M}_{\beta_2}[\varphi_v(P_t), \varphi_v(Q_t)]^T$
7: $K_P \leftarrow [2^{f-e}]([d_1]\varphi_u(P_s), P)$
8: $K_Q \leftarrow [2^{f-e}]([d_1]\varphi_u(Q_s), Q)$
9: $E \times E', [(P, P'), (Q, Q')] \leftarrow$ **Isogeny22Chain**$(K_P, K_Q, [(\varphi_u(P_s), 0_{E_v}), (\varphi_u(Q_s), 0_{E_v})])$
10: **if** $e_{2f}(P, Q) = e_{2f}(P_s, Q_s)^{u^2 d_1}$ **then** // We identify the right curve E_I by ensuring it is d_1-isogeneous to E_0
11: $E_I \leftarrow E$, $P_I \leftarrow P$, and $Q_I \leftarrow Q$
12: **else**
13: $E_I \leftarrow E'$, $P_I \leftarrow P'$, and $Q_I \leftarrow Q'$
14: $[P_I, Q_I]^T \leftarrow \left[\frac{1}{ud_1}\right] \mathbf{M}_{\beta_1}[P_I, Q_I]^T$
15: **end if**
16: **return** $E_I, (P_I, Q_I)$

Now we derive the bound for **IdealToIsogeny** from the bound of **SuitableIdeals**.
The algorithm **IdealToIsogeny** is as follows: (Algorithm 5)

Lemma 7. *In original Round-2 SQIsign, while computing **IdealToIsogeny** (Algorithm 5), the maximum size of the integers during the quaternion operations is less than*

$$2^{50} p^{10} (\log p)^{96} \mathrm{nrd}(I)^4.$$

Proof. Since the maximum value of the size of integers of **SuitableIdeals** is at most the stated bound, in **line 1**, it also holds.
$d_1 = q_I(\beta_1), d_2 = q_I(\beta_2)$ so it also holds for **line 2,3** by Lemma 6.

□

Bound on RandomIdealGivenNorm. Now we derive the bound for **RandomIdealGivenNorm**.

The algorithm **RandomIdealGivenNorm** is as follows: (Algorithm 6)

Algorithm 6 RandomIdealGivenNorm(N, prime)

Input: A positive integer N not multiple of p which is the norm of some left $\mathcal{O}_0$ ideal, A boolean *prime* indicating whether N is prime, and a fixed parameter $m = \texttt{QUAT_prime_cofactor}$ which is prime, approximately as large as p and larger than N but distinct from both.
Output: A random left ideal J' of O_0 of norm N, or raise an exception.

1: $found \leftarrow$ false
2: **if** prime **then**
3: **while** not $found$ **do**
4: $g_1, g_2, g_3 \leftarrow$ independent uniformly random integers in $[0, N-1]$
5: $\gamma \leftarrow g_1 i + g_2 j + g_3 ij$
6: $found \leftarrow \big(1 = \text{Legendre}(-\text{nrd}(\gamma), N)\big)$
7: **if** $found$ **then**
8: $\gamma \leftarrow \gamma + \text{ModularSQRT}(-\text{nrd}(\gamma), N)$
9: **end if**
10: **end while**
11: **else**
12: $\gamma \leftarrow$ **GeneralizedRepresentInteger**$(mN, i, O_0, \text{false})$ ▷ This might raise an exception
13: **end if**
14: **while** not $found$ **do**
15: $x, y, z, w \leftarrow$ uniformly randomly selected integers in $[1, N]$
16: $\beta \leftarrow x + yi + zj + wij$
17: $found \leftarrow \big(\gcd(\text{nrd}(\beta), N) = 1\big)$
18: **end while**
19: $J' \leftarrow$ the left O_0-ideal generated by $\gamma\beta$ and N
20: **return** J'

Lemma 8. *While executing* ***RandomIdealGivenNorm*** *(Algorithm 6), the maximum absolute value of all intermediate integers is less than*

$$32\, p\, N^2.$$

Proof. Let $m :=$ QUAT_prime_cofactor (See Appendix E).
Lines 4–5. The samples satisfy $g_1, g_2, g_3 \in [0, N-1]$, hence $|g_i| < N$.
Lines 6 and 8. Both **Legendre** and **ModularSQRT** operate modulo N, so their outputs are reduced modulo N and so the maximum size of coefficients of γ are less than $2N$.
Line 12. By Lemma 19, the call to **GeneralizedRepresentInteger** with input $M = mN$ yields integers of size at most $4mN - p$.
Lines 15–16. The samples $x, y, z, w \in [1, N]$ imply $|x|, |y|, |z|, |w| \le N$.
Line 19. Let J' be the left $\mathcal{O}_0$-ideal generated by $\gamma\beta$ and N. Since the output of **GeneralizedRepresentInteger** has coefficients of size at most $4mN - p$, the sizes of coefficients appearing in generators of J' are bounded by $2N(4mN - p)$. Finally, by Appendix E we have $m < 4p$. Hence

$$4mN \;<\; 4 \cdot (4p)\, N \;=\; 16\, p\, N,$$

Algorithm 7 KeyGen()

Parameter : The smallest prime D_{mix} larger than $2^{4\lambda}$
Output: Secret key sk and public key pk.

1: **while** true **do**
2: $I_{\text{sk}} \leftarrow$ **RandomIdealGivenNorm**$(D_{\text{mix}}, true)$
3: **try**
4: $I_{\text{sk}} \leftarrow$ **RandomEquivalentPrimeIdeal**(I_{sk})
5: E_{pk}, $\varphi_{\text{sk}}(P_0)$, $\varphi_{\text{sk}}(Q_0) \leftarrow$ **IdealToIsogeny**(I_{sk})
6: **except**
7: **continue**
8: P_{pk}, Q_{pk}, $hint_{\text{pk}} \leftarrow$ **TorsionBasisToHint**(E_{pk})
9: $M_{\text{sk}} \leftarrow$ **ChangeOfBasis**$_{2^f}\Big(E_{\text{pk}}, \big(\varphi_{\text{sk}}(P_0), \varphi_{\text{sk}}(Q_0)\big), \big(P_{\text{pk}}, Q_{\text{pk}}\big)\Big)$
10: $pk \leftarrow \big(E_{\text{pk}}, hint_{\text{pk}}\big)$
11: $sk \leftarrow \big(E_{\text{pk}}, hint_{\text{pk}}, I_{\text{sk}}, M_{\text{sk}}\big)$
12: **return** sk, pk
13: **end while**

so throughout **RandomIdealGivenNorm** the maximum absolute value of all intermediate integers is less than $2N \cdot 16pN = 32pN^2$, as claimed.

□

3.3 KeyGen and Sign

Combining bounds for the subroutines of **KeyGen** and **Sign** proven in Section 3.1, 3.2 and Appendix C, we derive the uniform worst-case bound for **KeyGen** and **Sign** in original Round-2 SQIsign.

The algorithm for the key generation of SQIsign is as follows: (Algorithm 7)

Remark 3. We propose a hypothesis to prove Theorem 1 and Theorem 2, which are statements about the uniform bound of the algorithms **KeyGen** and **Sign**, respectively. We heuristically claim that the probability that the output ideal of **RandomEquivalentPrimeIdeal** (Algorithm 12) has reduced norm more than p is negligible, by Appendix A.1 of [27]. Based on this claim, we assume our hypothesis that the reduced norm of I_{sk} is at most p from **line 4**.

Bound on KeyGen. Now we derive the bound for **KeyGen**, based on the bound for **IdealToIsogeny**.

Proposition 1 (Original KeyGen bound). *In original Round-2 SQIsign, let p be a parameter of SQIsign depending on the security parameter λ. While executing **KeyGen**, suppose that each input basis of ideal multiplication is δ-LLL-reduced for any arbitrary $\frac{3}{4} \leq \delta < 1$. Then, the maximum size of the integers during the operations over a quaternion algebra is less than*

$$2^{50} p^{14} (\log p)^{96}.$$

Proof. In **line 2**, by Lemma 8, the maximum size of integers in **RandomIdealGivenNorm** is less than $32p(D_{\mathrm{mix}})^2 \leq 32p(2^{8\lambda}+981)^2 \leq 2^{65}p^9$ (See Appendix E for the size bound of D_{mix}).

In **line 4**, by Lemma 20, integers representing I_{sk} become 64 times of them of $\hat{I}_{\mathrm{sk}}$, and actually they do not increase the size heuristically by [17]. We ignore the exceptional branch since the failure probability of **RandomEquivalentPrimeIdeal** is less than 2^{-64}.

In **line 5**, by Lemma 7, the maximum size of the integers is less than

$$2^{50}p^{10}(\log p)^{96}\mathrm{nrd}(I_{\mathrm{sk}})^4.$$

By the hypothesis that $\mathrm{nrd}(I_{\mathrm{sk}}) \leq p$ (we claim this hypothesis to bound the reduced norm of I_{sk}), we have the size bound as

$$2^{50}p^{14}(\log p)^{96}.$$

□

Bound on Sign. Now we derive the bound for **Sign**, by combining the bounds of subroutines.

The algorithm for the sign of SQIsign is as followed: (Algorithm 8)

Proposition 2 (Original Sign bound). *In original Round-2 SQIsign, let p be a parameter of SQIsign depending on the security parameter λ. While executing **Sign**, suppose that each input basis of ideal multiplication is δ-LLL-reduced for any arbitrary $\frac{3}{4} \leq \delta < 1$. Then, the maximum size of the integers during the operations over a quaternion algebra is less than*

$$2^{50}p^{95}.$$

Proof. In **line 4**, by Lemma 8, the maximum value of the size of integers is less than $32p(D_{\mathrm{mix}})^2 \leq 32p(2^{4\lambda}+981)^2 \leq 2^{65}p^9$ as in the proof of Theorem 1, and the output I_{com} has the reduced norm D_{mix}.

In **line 6**, by Lemma 20, the maximum value of the size of integers is less than 64 times of that representing $\hat{I}_{\mathrm{com}}$ and the output I_{com} has the reduced norm at most $\frac{2^{15}}{\pi^2}p$ by the same reason in the proof of Theorem 1.

In **line 7**, by Lemma 7, the maximum value of the size of integers is less than

$$2^{16}(\log p)^{24}D_{\mathrm{mix}}^{12} \leq 2^{16}(\log p)^{24}(2^{4\lambda}+643)^{12}.$$

In **line 10**, since we use SHAKE256 [17], the maximum size is less than $\sqrt{p}$.

In **line 11**, since M_{sk} is a change-of-basis matrix from $(\varphi_{\mathrm{sk}}(P_0), \varphi_{\mathrm{sk}}(Q_0))$ to B_{pk}, the maximum size of its entries is less than p. Hence, $c_1, c_2 < p\sqrt{p}$.

In **line 12**, by Lemma 23, the maximum size of integers is less than p.

In **line 13**, since $[I_{\mathrm{sk}}]_* I'_{\mathrm{chl}} = I_{\mathrm{sk}}^{-1}(I_{\mathrm{sk}} \cap I'_{\mathrm{chl}})$, where intersection is dual of the addition of dual ideals and inverse of I is $\frac{1}{\mathrm{nrd}(I)}\bar{I}$, we can see that during the computation, the maximum size of integers is less than

$$\begin{aligned}
&2^{10}p^{10}(2\mathrm{nrd}(I_{\mathrm{sk}})^2)^{32}2^8\mathrm{nrd}(I_{\mathrm{sk}})^{16}\mathrm{nrd}(I_{\mathrm{sk}} \cap I'_{\mathrm{chl}})^4 \\
&\leq 2^{50}p^{10}\mathrm{nrd}(I_{\mathrm{sk}})^{80}\mathrm{nrd}(I_{\mathrm{sk}} \cap I'_{\mathrm{chl}})^4 \leq 2^{50}p^{10}\mathrm{nrd}(I_{\mathrm{sk}})^{84}\mathrm{nrd}(I'_{\mathrm{chl}})^4 \\
&\leq 2^{50}p^{91}2^{4f} \leq 2^{50}p^{95}
\end{aligned}$$

by Lemma 2, Lemma 5, and Lemma 10, since $\mathrm{nrd}(I_{\mathrm{sk}}) \leq p$ and $\mathrm{nrd}(I'_{\mathrm{chl}}) \leq 2^f$.

In **line 14**, first we compute the multiplication $I_{\mathrm{sk}} \cdot I_{\mathrm{chl}}$, where

$$\mathrm{nrd}(I_{\mathrm{chl}}) = \mathrm{nrd}([I_{\mathrm{sk}}]_* I'_{\mathrm{chl}}) = \mathrm{nrd}\left(\mathrm{nrd}(I_{\mathrm{chl}})I_{\mathrm{chl}}^{-1}\right)\mathrm{nrd}(I'_{\mathrm{chl}}) = 1{\cdot}\mathrm{nrd}(I'_{\mathrm{chl}}) = \mathrm{nrd}(I'_{\mathrm{chl}})$$

so by Lemma 2 and Lemma 5, the maximum size of integers during the multiplication is less than

$$\begin{aligned}
&2^{10}p^{10}2^{32}(2\,\mathrm{nrd}(I_{\mathrm{sk}}))^8\mathrm{nrd}(I_{\mathrm{sk}})^{16}\mathrm{nrd}(I_{\mathrm{chl}})^4 \\
&= 2^{50}p^{10}\mathrm{nrd}(I_{\mathrm{sk}})^{24}\mathrm{nrd}(I'_{\mathrm{chl}})^4 \leq 2^{50}p^{34}2^{4f} \leq 2^{50}p^{38}.
\end{aligned}$$

By Lemma 24, while computing the algorithm **RandomEquivalentQuaternion**, the maximum size of integers is less than p^6.

From **line 16** to **line 20**, the maximum size of integers does not grow clearly. In **line 23**, by Lemma 8, the maximum size of integers is less than

$$32p(2^{e'_{\mathrm{rsp}}-q_{\mathrm{rsp}}})^2 < 32p \cdot 2^{2e'_{\mathrm{rsp}}} < 32p \cdot 2^{2e_{\mathrm{rsp}}} = 32p \cdot 2^{2\lceil \log_2 \sqrt{p} \rceil} < p^3.$$

In **line 24**, while computing $I_{\mathrm{com,rsp}} \cap I_{\mathrm{aux}}$, by Lemma 2 and Lemma 5, the maximum size of integers is less than

$$\begin{aligned}
&2^{10}p^{10}2^{32}2^8\mathrm{nrd}(I_{\mathrm{com,rsp}})^{16}\mathrm{nrd}(I_{\mathrm{aux}})^4 \\
&\leq 2^{50}p^{10}(q_{\mathrm{rsp}}D_{\mathrm{mix}})^{16}(2^{e'_{\mathrm{rsp}}} - q_{\mathrm{rsp}})^4 \leq 2^{50}p^{10}(D_{\mathrm{mix}} \cdot 2^{n_{\mathrm{bt}}})^{16}(2^{e_{\mathrm{rsp}}})^4 \\
&\leq 2^{50}p^{10}(2^8p^2)^{16}(2^{1/4}\sqrt{p})^4 \leq 2^{179}p^{44}
\end{aligned}$$

by the fact that

$$\begin{aligned}
\mathrm{nrd}(I_{\mathrm{com,rsp}}) &= \mathrm{nrd}(\mathcal{O}_0\alpha_{\mathrm{rsp}} + \mathcal{O}_0(q_{\mathrm{rsp}}D_{\mathrm{mix}})) = q_{\mathrm{rsp}}D_{\mathrm{mix}} \\
&\leq d_{\mathrm{rsp}}D_{\mathrm{mix}} = D_{\mathrm{rsp}} \cdot D_{\mathrm{mix}}^2 \cdot 2^f / D_{\mathrm{mix}}^2 \cdot 2^{f-n_{\mathrm{bt}}} = D_{\mathrm{rsp}} \cdot 2^{n_{\mathrm{bt}}},
\end{aligned}$$

$$\mathrm{nrd}(I_{\mathrm{aux}}) = 2^{e'_{\mathrm{rsp}}} - q_{\mathrm{rsp}} < 2^{e'_{\mathrm{rsp}}} \leq 2^{e_{\mathrm{rsp}}-1} < 2^{\log_2 \sqrt{p}} = \sqrt{p}.$$

While applying **IdealToIsogeny**, by Lemma 7, the maximum size of integers is

$$\begin{aligned}&2^{50}p^{10}(\log p)^{96}\mathrm{nrd}(I_{\mathrm{com,rsp}} \cap I_{\mathrm{aux}})^4\\ &\leq 2^{50}p^{10}(\log p)^{96}\mathrm{nrd}(I_{\mathrm{com,rsp}})^4\mathrm{nrd}(I_{\mathrm{aux}})^4\\ &\leq 2^{50}p^{10}(\log p)^{96}(D_{\mathrm{mix}}q_{\mathrm{rsp}})^4(2^{e'_{\mathrm{rsp}}} - q_{\mathrm{rsp}})^4 \leq 2^{83}p^{20}(\log p)^{96}.\end{aligned}$$

by Lemma 9 (stated later in this section).

In **line 30**, by Lemma 7, the maximum size of integers is less than

$$\begin{aligned}&2^{50}p^{10}(\log p)^{96}\mathrm{nrd}(I_{\mathrm{com,rsp}})^4 \leq 2^{50}p^{10}(\log p)^{96}(D_{\mathrm{mix}}q_{\mathrm{rsp}})^4\\ &\leq 2^{50}p^{10}(\log p)^{96}(2^8p^2)^4 = 2^{82}p^{18}(\log p)^{96}.\end{aligned}$$

In **line 36**, by Lemma 26 and Lemma 27, the maximum size of integers is at most

$$2^{r'_{\mathrm{rsp}}} \leq 2^{\log_2 d_{\mathrm{rsp}}} = d_{\mathrm{rsp}} < \mathrm{nrd}(\alpha_{\mathrm{rsp}}) < 4 \cdot D_{\mathrm{rsp}} \cdot D_{\mathrm{mix}}^2 \cdot 2^f < p^5.$$

□

Lemma 9. *Let $\mathcal{O}$ be a maximal order in $B_{p,\infty}$ and I_1, I_2 be intersection of two left $\mathcal{O}$-ideals. Then,*

$$\mathrm{nrd}(I_1 \cap I_2) \leq \mathrm{nrd}(I_1)\mathrm{nrd}(I_2).$$

Proof. See Appendix A.3. □

Lemma 10. *Let $\mathcal{O}$ be a maximal order in $B_{p,\infty}$, and let I_1, I_2 be left $\mathcal{O}$-ideals. Fix $\mathbb{Z}$-bases $I_1 = \langle \alpha_1, \ldots, \alpha_4 \rangle$ and $I_2 = \langle \beta_1, \ldots, \beta_4 \rangle$. Define*

$$r_1 := \mathrm{lcm}(r_{\alpha_1}, r_{\alpha_2}, r_{\alpha_3}, r_{\alpha_4}), \qquad r_2 := \mathrm{lcm}(r_{\beta_1}, r_{\beta_2}, r_{\beta_3}, r_{\beta_4}),$$

and form the integer matrices

$$M_1 := [\, r_1\alpha_1 \ \cdots \ r_1\alpha_4 \,] \in \mathbb{Z}^{4\times 4}, \qquad M_2 := [\, r_2\beta_1 \ \cdots \ r_2\beta_4 \,] \in \mathbb{Z}^{4\times 4},$$

where columns are the coefficient vectors in the fixed basis $(1, i, j, k)$. Let $M := (M_1 \mid M_2) \in \mathbb{Z}^{4\times 8}$. [1] *Then*

$$\gcd\{\det(A) : A \text{ is a } 4\times 4 \text{ submatrix of } M\} \leq \min\left(r_1^4\,\mathrm{nrd}(I_1)^2, r_2^4\,\mathrm{nrd}(I_2)^2\right).$$

Proof. See Appendix A.4. □

[1] This process is the addition of two ideals. See 3.1.5.2 of [17].

Algorithm 8 Sign(sk, msg)

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

Parse sk as $(E_{\mathrm{pk}},\ hint_{\mathrm{pk}},\ I_{\mathrm{sk}},\ M_{\mathrm{sk}})$
$P_{\mathrm{pk}}, Q_{\mathrm{pk}} \leftarrow$ **TorsionBasisFromHint**$(E_{\mathrm{pk}}, hint_{\mathrm{pk}})$
while true **do**
  // Commitment
  $I_{\mathrm{com}} \leftarrow$ **RandomIdealGivenNorm**$(D_{\mathrm{mix}}, true)$
  try
    $I_{\mathrm{com}} \leftarrow$ **RandomEquivalentPrimeIdeal**(I_{com})
    $E_{\mathrm{com}}, P_{\mathrm{com}}, Q_{\mathrm{com}} \leftarrow$ **IdealToIsogeny**(I_{com})
  except
    continue
  // Challenge
  $chl \leftarrow \mathrm{HASH}\big(pk \,\|\, j(E_{\mathrm{com}}) \,\|\, msg\big)$
  // Response
  $(c_1, c_2) \leftarrow M_{\mathrm{sk}} \cdot (1, chl)$
  $I'_{\mathrm{chl}} \leftarrow$ **KernelDecomposedToIdeal**$_{2^f}(c_1, c_2)$
  $I_{\mathrm{chl}} \leftarrow [I_{\mathrm{sk}}]_* I'_{\mathrm{chl}}$ *// I_{chl} is the ideal corresponding to φ_{chl}*
  $\alpha_{\mathrm{rsp}} \leftarrow$ **RandomEquivalentQuaternion**$(I_{\mathrm{com}} \cap (I_{\mathrm{sk}} \cdot I_{\mathrm{chl}}))$
  $\alpha_{\mathrm{rsp}},\ n_{\mathrm{bt}} \leftarrow$ **ComputeBacktrackingAndNormalize**(α_{rsp})
  $d_{\mathrm{rsp}} \leftarrow \mathrm{nrd}(\alpha_{\mathrm{rsp}})/D_{\mathrm{mix}}^2 \cdot 2^{f-n_{\mathrm{bt}}}$ *// d_{rsp} is the degree of φ_{rsp}*
  $r_{\mathrm{rsp}} \leftarrow$ DyadicValuation(d_{rsp})
  $q_{\mathrm{rsp}} \leftarrow d_{\mathrm{rsp}}/2^{r_{\mathrm{rsp}}}$
  $I_{\mathrm{com,rsp}} \leftarrow \mathcal{O}\alpha_{\mathrm{rsp}} + \mathcal{O}(q_{\mathrm{rsp}} D_{\mathrm{mix}})$ *// $I_{\mathrm{com,rsp}}$ corresponds to $\varphi_{\mathrm{com}} \circ \varphi_{\mathrm{rsp}}$*
  $e'_{\mathrm{rsp}} \leftarrow e_{\mathrm{rsp}} - r_{\mathrm{rsp}} - n_{\mathrm{bt}}$ *// e'_{rsp} is the degree of the two-dimensional isogeny*
  if $e'_{\mathrm{rsp}} > 0$ **then**
    try
      $I_{\mathrm{aux}} \leftarrow$ **RandomIdealGivenNorm**$(2^{e'_{\mathrm{rsp}}} - q_{\mathrm{rsp}}, false)$
      $E'_{\mathrm{aux}}, P'_{\mathrm{aux}}, Q'_{\mathrm{aux}} \leftarrow$ IdealToIsogeny$(I_{\mathrm{com,rsp}} \cap I_{\mathrm{aux}})$
    except
      continue
    $(E_{\mathrm{aux}}, P_{\mathrm{aux}}, Q_{\mathrm{aux}}, E_{\mathrm{chl}}, P_{\mathrm{chl}}, Q_{\mathrm{chl}}) \leftarrow$ **SplitAuxiliaryIsogeny**
    $(E_{\mathrm{com}}, E'_{\mathrm{aux}}, P_{\mathrm{com}}, Q_{\mathrm{com}}, P'_{\mathrm{aux}}, Q'_{\mathrm{aux}}, q_{\mathrm{rsp}}, e'_{\mathrm{rsp}}, r_{\mathrm{rsp}}, n_{\mathrm{bt}})$
  else
    try
      $E_{\mathrm{chl}}, P_{\mathrm{chl}}, Q_{\mathrm{chl}} \leftarrow$ **IdealToIsogeny**$(I_{\mathrm{com,rsp}})$
      $E_{\mathrm{aux}}, P_{\mathrm{aux}}, Q_{\mathrm{aux}} \leftarrow E_{\mathrm{chl}}, P_{\mathrm{chl}}, Q_{\mathrm{chl}}$
    except
      continue
  end if
  if $r_{\mathrm{rsp}} > 0$ **then**
    $E_{\mathrm{chl}}, P_{\mathrm{chl}}, Q_{\mathrm{chl}} \leftarrow$ **ComputeEvenNonBacktrackingResponse**
    $(E_{\mathrm{chl}}, P_{\mathrm{chl}}, Q_{\mathrm{chl}}, \alpha_{\mathrm{rsp}}, e'_{\mathrm{rsp}}, r_{\mathrm{rsp}})$
  end if
  $E_{\mathrm{chl}}, P_{\mathrm{chl}}, Q_{\mathrm{chl}} \leftarrow$ **ComputeChallengeIsogeny**
  $(E_{\mathrm{pk}}, chl, P_{\mathrm{pk}}, Q_{\mathrm{pk}}, E_{\mathrm{chl}}, P_{\mathrm{chl}}, Q_{\mathrm{chl}}, n_{\mathrm{bt}})$
  $M_{\mathrm{chl}},\ hint_{\mathrm{aux}},\ hint_{\mathrm{chl}} \leftarrow$ **SetChangeOfBasisMatrix**
  $(E_{\mathrm{aux}}, E_{\mathrm{chl}}, P_{\mathrm{aux}}, Q_{\mathrm{aux}}, P_{\mathrm{chl}}, Q_{\mathrm{chl}}, e'_{\mathrm{rsp}} + r_{\mathrm{rsp}})$
  $\sigma \leftarrow (E_{\mathrm{aux}}, n_{\mathrm{bt}}, r_{\mathrm{rsp}}, M_{\mathrm{chl}}, chl, hint_{\mathrm{aux}}, hint_{\mathrm{chl}})$
  return σ
end while

4 The Uniform Bound of SQIsign with Modified IdealMultiplication

In this section, we propose a modified algorithm to compute the multiplication of two ideals with a smaller size for fixed-precision arithmetic. Based on this modification, we compute a new uniform worst-case bound for SQIsign, which is meaningfully smaller than the one we computed in Section 3. (See Table 2)

4.1 Modified IdealMultiplication

Let $I_1 = \langle \alpha_1, \alpha_2, \alpha_3, \alpha_4 \rangle$, $I_2 = \langle \beta_1, \beta_2, \beta_3, \beta_4 \rangle$ be ideals and our goal is to multiply these two ideals with small size fixed-precision arithmetic. The original version of ideal multiplication in SQIsign uses modulus m for HNF algorithm, as a determinant of a matrix whose columns are $\alpha_1\beta_j$ $(1 \leq j \leq 4)$. We propose a modified algorithm to reduce the uniform bound. (See Algorithm 9) We set $m := \mathrm{nrd}(r_1 I_1)\mathrm{nrd}(r_2 I_2)$ where r_1, r_2 are denominators of I_1, I_2, respectively. It lowers the resulting worst-case bound by decreasing the size of m used for computing HNF over $\mathbb{Z}_m$. (Compare Lemma 2 and Lemma 12)

Modification of IdealMultiplication. Before giving the algorithm, we justify that this modulus m is valid, comparing it with the greatest common divisor of all 4×4 minors, in Lemma 11.

Lemma 11. *Let I_1, I_2 be left ideals of maximal orders $\mathcal{O}_1, \mathcal{O}_2$ in $B_{p,\infty}$ such that $O_R(I_1) = O_L(I_2)$ with $r_i := \min\{r \in \mathbb{Z} \mid rI_i \subset \mathbb{Z}^4\}$, respectively, and let M be a 4×16 matrix where $M_{1,l}, M_{2,l}, M_{3,l}, M_{4,l}$ are coefficients of $r_1 r_2 \alpha_{\lceil \frac{l}{4} \rceil} \beta_{(l \bmod 4)+1}$ for $1, i, j, k$, respectively. Then,*

$$\gcd\{\det(A) \mid A \text{ is a } 4 \times 4 \text{ submatrix of } M\} = \mathrm{nrd}(r_1 I_1)^2 \mathrm{nrd}(r_2 I_2)^2.$$

Proof. See Appendix A.5. □

From the correctness proven by Lemma 11, we propose the modified **IdealMultiplication** algorithm as follows: (Algorithm 9)

Bound on modified IdealMultiplication. We derive a size bound of modified **IdealMultiplication**, which is dramatically smaller than the one of **IdealMultiplication** of the original Round-2 SQIsign, given in Lemma 2.

Lemma 12. *In the algorithm **IdealMultiplication**, the maximum size of the integers is less than*

$$r_1^8 r_2^8 \,\mathrm{nrd}(I_1)^4 \mathrm{nrd}(I_2)^4.$$

Algorithm 9 IdealMultiplication(I_1, I_2) (Modified Version)

Input: A left $\mathcal{O}_1$-ideal $I_1 = \langle \alpha_1, \alpha_2, \alpha_3, \alpha_4 \rangle$ and a left $\mathcal{O}_2$-ideal $I_2 = \langle \beta_1, \beta_2, \beta_3, \beta_4 \rangle$ such that $O_R(I_1) = O_L(I_2)$.
Output: A left $\mathcal{O}_1$-ideal $I_1 I_2 = \langle \gamma_1, \gamma_2, \gamma_3, \gamma_4 \rangle$.
1: $r_1 \leftarrow \mathrm{lcm}(r_{\alpha_1}, r_{\alpha_2}, r_{\alpha_3}, r_{\alpha_4})$, $r_2 \leftarrow \mathrm{lcm}(r_{\beta_1}, r_{\beta_2}, r_{\beta_3}, r_{\beta_4})$
2: $\alpha_i \leftarrow r_1 \alpha_i$, $\beta_j \leftarrow r_2 \beta_j$ for $1 \le i, j \le 4$
3: $m \leftarrow \mathrm{nrd}(r_1 I_1)^2 \mathrm{nrd}(r_2 I_2)^2$
4: Compute $M \leftarrow \mathbf{HNF}(\alpha_i \beta_j)_{1 \le i,j \le 4}$ over $\mathbb{Z}_m$.
5: **return** $\frac{1}{r_1 r_2} M$

Proof. In **line 3**, by Lemma 11, we have

$$m = \mathrm{nrd}(r_1 I_1)^2 \mathrm{nrd}(r_2 I_2)^2 = r_1^4 r_2^4 \, \mathrm{nrd}(I_1)^2 \mathrm{nrd}(I_2)^2.$$

In **line 4**, since **HNF** consists of addition, subtraction, multiplication, and division, the maximum size of integers is less than m^2 [25], which is at most

$$r_1^8 r_2^8 \, \mathrm{nrd}(I_1)^4 \mathrm{nrd}(I_2)^4.$$

□

4.2 IdealToIsogeny

We need to compute the new bound of **IdealToIsogeny**, since it needs to compute the multiplication of two ideals, in the subroutine **SuitableIdeals**. (See Algorithm 4)

So, we first recompute the bound of **SuitableIdeals** with our modified **IdealMultiplication**. We would derive the bound for **IdealToIsogeny** directly from this result.

Lemma 13. *In the algorithm **SuitableIdeals**, the maximum size of the integers is less than*

$$\max\left(2^{16}(\log p)^{16} \mathrm{nrd}(I)^4, \frac{64}{\pi^4} p^2\right).$$

Proof. First, when computing a Minkowski reduced basis, we compute the basis $\bar{J}_t I$ and the Gram matrix of it, then we compute **L2** of them.
Since each basis element of J_t has reduced norm less than $(\log p)^2$ by Appendix E, we can see that computing Minkowski reduced basis, the maximum size of integers is less than

$$2^{16}(\log p)^{16} \mathrm{nrd}(I)^4$$

by Lemma 12 and Lemma 5.

Other lines are same as the proof of Lemma 6.

□

Remark 4. One can replace **SuitableIdeals** with **Qlapoti** [28], as a subroutine of **IdealToIsogeny**. This replacement does not change the uniform bound. (See Lemma 29 of Appendix D.)

We can deduce the following lemma as a corollary of Lemma 13, by the same process of Lemma 7.

Lemma 14. *In **IdealToIsogeny**, the maximum value of the size of integers during the quaternion operations is less than*

$$\max\left(2^{16}(\log p)^{16}\mathrm{nrd}(I)^4, \frac{64}{\pi^4}p^2\right).$$

4.3 KeyGen and Sign

We now combine the uniform bounds for **KeyGen** and **Sign** with modified **IdealMultiplication**, from the results in Section 4.1, 4.2, and Appendix C. From these bounds, we obtain the uniform worst-case bound for **KeyGen** and **Sign** processes of SQIsign under the modified ideal multiplication.

Bound on KeyGen. By the similar procedure of Section 3.3, from the bound for **IdealToIsogeny** with modified **IdealMultiplication**, we derive the uniform bound of **KeyGen**.

Theorem 1 (Modified KeyGen bound). *In SQIsign with modified **Ideal-Multiplication**, let p be a parameter of SQIsign depending on the security parameter λ. While executing **KeyGen**, the maximum value of the size of integers during the operations over a quaternion algebra is less than*

$$2^{68}p^9.$$

Proof. In **line 2**, by Lemma 8 and [17], the maximum size of integers in **RandomIdealGivenNorm** is less than

$$32p(D_{\mathrm{mix}})^2 \leq 32p(2^{8\lambda}+981)^2 \leq 2^{68}p^9.$$

In **line 4**, by Lemma 20, integers representing I_{sk} become 64 times of them of $\hat{I}_{\mathrm{sk}}$, and actually they do not increase the size heuristically by [17]. We ignore the except occurs because the failure probability of **RandomEquivalentPrimeIdeal** is less than 2^{-64}.

In **line 5**, by Lemma 14, the maximum size of the integers is less than

$$2^{16}(\log p)^{16}\mathrm{nrd}(I_{\mathrm{sk}})^4.$$

Since $\mathrm{nrd}(I_{\mathrm{sk}}) \leq p$ by the assumption, we have the maximum size

$$2^{16}(\log p)^{16}p^4.$$

□

Bound on Sign. Now we derive the bound for **Sign** combining the bounds of subroutines, by the similar procedure of Section 3.3.

Theorem 2 (Modified Sign bound). *In SQIsign with modified **IdealMultiplication**, let p be a parameter of SQIsign depending on the security parameter λ. While executing **Sign**, the maximum value of the size of integers during the operations over a quaternion algebra is less than*

$$2^{16}p^{28}.$$

Proof. From **line 4** to **line 12**, it is same as the proof of Proposition 2.

In **line 13**, since $[I_{\mathrm{sk}}]_* I'_{\mathrm{chl}} = I_{\mathrm{sk}}^{-1}(I_{\mathrm{sk}} \cap I'_{\mathrm{chl}})$, where intersection is dual of the addition of dual ideals and inverse of I is $\frac{1}{\mathrm{nrd}(I)}\bar{I}$, we can see that during the computation, the maximum size of integers is less than

$$\begin{aligned}&(2\,\mathrm{nrd}(I_{\mathrm{sk}})^2)^8\, 2^8 \mathrm{nrd}(I_{\mathrm{sk}})^4 \mathrm{nrd}(I_{\mathrm{sk}} \cap I'_{\mathrm{chl}})^4 \le 2^{16}\mathrm{nrd}(I_{\mathrm{sk}})^{20}\mathrm{nrd}(I_{\mathrm{sk}} \cap I'_{\mathrm{chl}})^4 \\ &\le 2^{16}\mathrm{nrd}(I_{\mathrm{sk}})^{24}\mathrm{nrd}(I'_{\mathrm{chl}})^4 \le 2^{16}p^{24}2^{4f} \le 2^{16}p^{28}\end{aligned}$$

by Lemma 12, Lemma 5, and Lemma 10.

In **line 14**, first we compute the multiplication $I_{\mathrm{sk}} \cdot I_{\mathrm{chl}}$, where

$$\mathrm{nrd}(I_{\mathrm{chl}}) = \mathrm{nrd}([I_{\mathrm{sk}}]_* I'_{\mathrm{chl}}) = \mathrm{nrd}\left(\mathrm{nrd}(I_{\mathrm{chl}})I_{\mathrm{chl}}^{-1}\right)\mathrm{nrd}(I'_{\mathrm{chl}}) = 1{\cdot}\mathrm{nrd}(I'_{\mathrm{chl}}) = \mathrm{nrd}(I'_{\mathrm{chl}})$$

so by Lemma 12 and Lemma 5, the maximum size of integers during the multiplication is less than

$$\begin{aligned}2^8(2\,\mathrm{nrd}(I_{\mathrm{sk}}))^8\mathrm{nrd}(I_{\mathrm{sk}})^4\mathrm{nrd}(I_{\mathrm{chl}})^4 &= 2^{16}\mathrm{nrd}(I_{\mathrm{sk}})^{12}\mathrm{nrd}(I'_{\mathrm{chl}})^4 \\ &\le 2^{16}p^{12}2^{4f} \le 2^{16}p^{16}.\end{aligned}$$

By Lemma 24, while computing the algorithm **RandomEquivalentQuaternion**, the maximum size of integers is less than p^6.

In **line 15**, by Lemma 26, the maximum size of integers is at most

$$2\alpha_{\mathrm{rsp}} < 2\sqrt{D_{\mathrm{rsp}} \cdot D_{\mathrm{mix}}^2 \cdot 2^f} < p^3.$$

From **line 16** to **line 23**, it is same as the proof of Proposition 2.

In **line 24**, while computing $I_{\mathrm{com,rsp}} \cap I_{\mathrm{aux}}$, by Lemma 12 and Lemma 5, the maximum size of integers is less than

$$\begin{aligned}&2^8 \cdot 2^8\mathrm{nrd}(I_{\mathrm{com,rsp}})^4\mathrm{nrd}(I_{\mathrm{aux}})^4 \le 2^{16}(q_{\mathrm{rsp}}D_{\mathrm{mix}})^4(2^{e'_{\mathrm{rsp}}} - q_{\mathrm{rsp}})^4 \\ &\le 2^{16}(D_{\mathrm{mix}} \cdot 2^{n_{\mathrm{bt}}})^4(2^{e_{\mathrm{rsp}}})^4 \le 2^{16}(2^8p^2)^4(2^{1/4}\sqrt{p})^4 \le 2^{49}p^{12}\end{aligned}$$

since

$$\begin{aligned}\mathrm{nrd}(I_{\mathrm{com,rsp}}) &= \mathrm{nrd}(\mathcal{O}_0\alpha_{\mathrm{rsp}} + \mathcal{O}_0(q_{\mathrm{rsp}}D_{\mathrm{mix}})) = q_{\mathrm{rsp}}D_{\mathrm{mix}} \\ &\leq d_{\mathrm{rsp}}D_{\mathrm{mix}} = D_{\mathrm{rsp}} \cdot D_{\mathrm{mix}}^2 \cdot 2^f / D_{\mathrm{mix}}^2 \cdot 2^{f-n_{\mathrm{bt}}} = D_{\mathrm{rsp}} \cdot 2^{n_{\mathrm{bt}}}\end{aligned}$$

and

$$\mathrm{nrd}(I_{\mathrm{aux}}) = 2^{e'_{\mathrm{rsp}}} - q_{\mathrm{rsp}} < 2^{e'_{\mathrm{rsp}}} \leq 2^{e_{\mathrm{rsp}}-1} < 2^{\log_2 \sqrt{p}} = \sqrt{p}.$$

While applying **IdealToIsogeny**, by Lemma 14, the maximum size of integers is

$$\begin{aligned}&2^{16}(\log p)^{16}\mathrm{nrd}(I_{\mathrm{com,rsp}} \cap I_{\mathrm{aux}})^4 \leq 2^{16}(\log p)^{16}\mathrm{nrd}(I_{\mathrm{com,rsp}})^4\mathrm{nrd}(I_{\mathrm{aux}})^4 \\ &\leq 2^{16}(\log p)^{16}(D_{\mathrm{mix}}q_{\mathrm{rsp}})^4(2^{e'_{\mathrm{rsp}}} - q_{\mathrm{rsp}})^4 \leq 2^{49}(\log p)^{16}p^{12}\end{aligned}$$

by Lemma 9.

In **line 30**, by Lemma 14, the maximum size of integers is less than

$$\begin{aligned}&2^{16}(\log p)^{16}\mathrm{nrd}(I_{\mathrm{com,rsp}})^4 \leq 2^{16}(\log p)^{16}(D_{\mathrm{mix}}q_{\mathrm{rsp}})^4 \\ &\leq 2^{16}(\log p)^{16}(2^8p^2)^4 = 2^{48}(\log p)^{16}p^8.\end{aligned}$$

In **line 36**, the proof is the same with Proposition 2.

□

Acknowledgements. This work was supported by Korea University grants K2511691 and K2511701.

References

1. Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM. In *2018 IEEE European symposium on security and privacy (EuroS&P)*, pages 353–367. IEEE, 2018.
2. Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 238–268, 2018.
3. Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, Zhenfei Zhang, et al. Falcon: Fast-Fourier lattice-based compact signatures over NTRU. *Submission to the NIST's post-quantum cryptography standardization process*, 36(5):1–75, 2018.
4. Daniel J Bernstein, Andreas Hülsing, Stefan Kölbl, Ruben Niederhagen, Joost Rijneveld, and Peter Schwabe. The SPHINCS+ signature framework. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pages 2129–2146, 2019.
5. Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loıc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Edoardo Persichetti, Gilles Zémor, and IC Bourges. Hamming quasi-cyclic (HQC). *NIST PQC Round*, 2(4):13, 2018.
6. Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. SQISign: compact post-quantum signatures from quaternions and isogenies. In *International conference on the theory and application of cryptology and information security*, pages 64–93. Springer, 2020.
7. Max Deuring. Die typen der multiplikatorenringe elliptischer funktionenkörper. In *Abhandlungen aus dem mathematischen Seminar der Universität Hamburg*, volume 14, pages 197–272. Springer Berlin/Heidelberg, 1941.
8. Markus Sosnowski, Florian Wiedner, Eric Hauser, Lion Steger, Dimitrios Schoinianakis, Sebastian Gallenmüller, and Georg Carle. The performance of post-quantum tls 1.3. In *Companion of the 19th International Conference on emerging Networking EXperiments and Technologies*, pages 19–27, 2023.
9. Filip Forsby, Martin Furuhed, Panos Papadimitratos, and Shahid Raza. Lightweight x. 509 digital certificates for the internet of things. In *International Conference on Interoperability in IoT*, pages 123–133. Springer, 2017.
10. Gabriele Restuccia, Hannes Tschofenig, and Emmanuel Baccelli. Low-power IoT communication security: On the performance of DTLS and TLS 1.3. In *2020 9th IFIP International Conference on Performance Evaluation and Modeling in Wireless Networks (PEMWN)*, pages 1–6. IEEE, 2020.
11. Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 423–447. Springer, 2023.
12. Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 448–471. Springer, 2023.
13. Damien Robert. Breaking SIDH in polynomial time. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 472–503. Springer, 2023.

14. Andrea Basso, Pierrick Dartois, Luca De Feo, Antonin Leroux, Luciano Maino, Giacomo Pope, Damien Robert, and Benjamin Wesolowski. SQIsign2D–west: the fast, the small, and the safer. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 339–370. Springer, 2024.
15. Marius A. Aardal, Andrea Basso, Luca De Feo, Sikhar Patranabis, and Benjamin Wesolowski. A complete security proof of SQIsign. In *Annual International Cryptology Conference*, pages 190–222. Springer, 2025.
16. Torbjörn Granlund. Gnu mp. *The GNU Multiple Precision Arithmetic Library*, 2(2), 1996.
17. 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.
18. John Voight. *Quaternion algebras*. Springer Nature, 2021.
19. Aurel Page and Benjamin Wesolowski. The supersingular endomorphism ring and one endomorphism problems are equivalent. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 388–417. Springer, 2024.
20. Benjamin Wesolowski. The supersingular isogeny path and endomorphism ring problems are equivalent. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1100–1111. IEEE, 2022.
21. Sina Schaeffler. Algorithms for Quaternion Algebras in SQIsign. 2023.
22. Paul D Domich, Ravindran Kannan, and Leslie E Trotter Jr. Hermite normal form computation using modulo determinant arithmetic. *Mathematics of operations research*, 12(1):50–59, 1987.
23. Arne Storjohann and George Labahn. Asymptotically fast computation of Hermite normal forms of integer matrices. In *Proceedings of the 1996 international symposium on Symbolic and algebraic computation*, pages 259–266, 1996.
24. David Eberly. The Laplace expansion theorem: Computing the determinants and inverses of matrices. *Geometric Tools, LLC, Scottsdale, Ariz, USA*, 2007.
25. Darrel Hankerson, Scott Vanstone, and Alfred Menezes. *Guide to elliptic curve cryptography*. Springer, 2004.
26. Pierrick Dartois, Antonin Leroux, Damien Robert, and Benjamin Wesolowski. SQISignHD: new dimensions in cryptography. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 3–32. Springer, 2024.
27. David Kohel, Kristin Lauter, Christophe Petit, and Jean-Pierre Tignol. On the quaternion ℓ-isogeny path problem. *LMS Journal of Computation and Mathematics*, 17(A):418–432, 2014.
28. 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 *International Conference on the Theory and Application of Cryptology and Information Security*, pages 174–205. Springer, 2025.
29. Viakalathur Shankar Sunder et al. *Operators on Hilbert space*, volume 71. Springer, 2016.

30. Rajendra Bhatia. Positive definite matrices. In *Positive Definite Matrices*. Princeton university press, 2009.
31. Daniele Micciancio and Shafi Goldwasser. *Complexity of lattice problems: a cryptographic perspective*, volume 671. Springer Science & Business Media, 2002.
32. David Steven Dummit, Richard M Foote, et al. *Abstract algebra*, volume 3. Wiley Hoboken, 2004.
33. Charles A Weibel. *An introduction to homological algebra*. Number 38. Cambridge university press, 1994.
34. Michael F Atiyah and Ian Grant Macdonald. *Introduction to commutative algebra*. CRC Press, 2018.
35. Henri Cohen. *A course in computational algebraic number theory*, volume 138. Springer Science & Business Media, 2013.
36. Phong Q Nguyen and Damien Stehlé. An LLL algorithm with quadratic complexity. *SIAM Journal on Computing*, 39(3):874–903, 2009.
37. Victor Shoup. *A computational introduction to number theory and algebra*. Cambridge university press, 2009.
38. Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.
39. Arnold Pizer. An algorithm for computing modular forms on Γ_0 (N). *Journal of algebra*, 64(2):340–390, 1980.

A Additional Proofs

In this section, we provide proofs of some lemmas used in this paper.

A.1 Proof of Lemma 1

For $x = (a_1, b_1, c_1, d_1)$ and $y = (a_2, b_2, c_2, d_2)$ in $\mathbb{R}^4$, define an inner product and a norm induced by the inner product

$$\langle x, y\rangle_{\mathbb{Z}} := a_1a_2 + b_1b_2 + p\,c_1c_2 + p\,d_1d_2 \qquad \text{and} \qquad \|x\|^2_{\mathbb{Z}} := \langle x, x\rangle_{\mathbb{Z}}.$$

Define $v(\gamma) := (a_\gamma, b_\gamma, c_\gamma, d_\gamma) \in \mathbb{R}^4$ denoting the coordinate vector of $\gamma \in B_{p,\infty}$, then $\|v(\gamma)\|^2_{\mathbb{Z}} = \mathrm{nrd}(\gamma)$. Let $s = \lceil l/4 \rceil$ and $t = (l-1) \bmod 4 + 1$. Using the multiplication rules in $B_{p,\infty}$, one checks that

$$\alpha_s\beta_t = (A_{s,t}) \cdot 1 + (B_{s,t}) \cdot i + (C_{s,t}) \cdot j + (D_{s,t}) \cdot k,$$

where

$$\begin{aligned}
A_{s,t} &= a_{\alpha_s}a_{\beta_t} - b_{\alpha_s}b_{\beta_t} - p\,c_{\alpha_s}c_{\beta_t} - p\,d_{\alpha_s}d_{\beta_t},\\
B_{s,t} &= a_{\alpha_s}b_{\beta_t} + b_{\alpha_s}a_{\beta_t} + p\,c_{\alpha_s}d_{\beta_t} - p\,d_{\alpha_s}c_{\beta_t},\\
C_{s,t} &= a_{\alpha_s}c_{\beta_t} - b_{\alpha_s}d_{\beta_t} + c_{\alpha_s}a_{\beta_t} + d_{\alpha_s}b_{\beta_t},\\
D_{s,t} &= a_{\alpha_s}d_{\beta_t} + b_{\alpha_s}c_{\beta_t} - c_{\alpha_s}b_{\beta_t} + d_{\alpha_s}a_{\beta_t}.
\end{aligned}$$

Thus $M_{1,l} = A_{s,t}$, $M_{2,l} = B_{s,t}$, $M_{3,l} = C_{s,t}$, and $M_{4,l} = D_{s,t}$.

Since $\mathrm{nrd}(a + bi + cj + dk) = \langle (a, b, c, d), (a, b, c, d)\rangle_{\mathbb{Z}}$,

$$|M_{1,l}| \le \sqrt{\mathrm{nrd}\left(\alpha_{\lceil \frac{l}{4} \rceil}\right) \mathrm{nrd}\left(\beta_{(l-1) \bmod 4+1}\right)}$$

by Cauchy–Schwarz inequality.

It is convenient to encode these coefficients by 4×4 real matrices B_r ($r = 1, 2, 3, 4$) so that $M_{r,\ell} = v(\alpha_i)^T B_r v(\beta_j)$. Explicitly,

$B_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & p & 0 \\ 0 & 0 & 0 & p \end{pmatrix}$, we can see that $\langle x, y\rangle_{\mathbb{Z}} = x^T B_1 y, \forall x, y \in \mathbb{Z}^4$.

To compute other norms corresponding to $M_{2,l}, M_{3,l}, M_{4,l}$, respectively, let

$$B_2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & p \\ 0 & 0 & p & 0 \end{pmatrix}, \quad B_3 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \quad B_4 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

Define a bilinear form (not an inner product)

$$\langle (a_1, b_1, c_1, d_1), (a_2, b_2, c_2, d_2)\rangle_i := a_1b_2 + b_1a_2 + pc_1d_2 + pd_1c_2.$$

Then,

$$\langle x, y\rangle_i = x^T B_2 y = \langle x, B_1^{-1} B_2 y\rangle_{\mathbb{Z}}, \forall x, y \in \mathbb{Z}^4$$

$$\because \langle x, B_1^{-1} B_2 y\rangle_{\mathbb{Z}} = x^T B_1 (B_1^{-1} B_2 y) = x^T B_2 y = \langle x, y\rangle_i.$$

$$\therefore \langle x, x\rangle_i = \langle x, B_1^{-1} B_2 x\rangle_{\mathbb{Z}} = \|x\|_{\mathbb{Z}} \|B_1^{-1} B_2 x\|_{\mathbb{Z}} \leq \|x\|_{\mathbb{Z}}^2 \sup_{\|u\|=1} \|B_1^{-1} B_2 u\|_{\mathbb{Z}}$$

$$\leq \sup_{\|u\|=1} \|B_1^{-1} B_2 u\|_{\mathbb{Z}} \cdot \sqrt{\mathrm{nrd}\left(\alpha_{\lceil \frac{l}{4} \rceil}\right) \mathrm{nrd}(\beta_{(l-1) \bmod 4+1})}$$

by Cauchy-Schwarz inequality.

$$\sup_{\|u\|=1} \|B_1^{-1} B_2 u\|_{\mathbb{Z}} = \sup_{\|v\|=1} \|\sqrt{B_1}(B_1^{-1} B_2)(\sqrt{B_1})^{-1} v\|_2$$

by Exercise 1.7.7 of [29], since B_1 is positive definite so

$$\langle x, y\rangle_{\mathbb{Z}} = x^T B_1 y = \langle x, B_1 y\rangle_2 = \langle \sqrt{B_1} x, \sqrt{B_1} y\rangle_2$$

by the proof of 1.1.(vi) of [30] hence a map $\sqrt{B_1} : x \mapsto \sqrt{B_1} x$ is an isometry from $(\mathbb{R}^4, \langle \cdot, \cdot \rangle_{\mathbb{Z}})$ to $(\mathbb{R}^4, \langle \cdot, \cdot \rangle_2)$.
The value $\sup_{\|v\|=1} \|\sqrt{B_1}(B_1^{-1} B_2)(\sqrt{B_1})^{-1} v\|_2$ is the maximum singular value of $\sqrt{B_1}(B_1^{-1} B_2)(\sqrt{B_1})^{-1}$.

Since each singular value is 1, we have

$$|M_{2,l}| \leq \sqrt{\mathrm{nrd}\left(\alpha_{\lceil \frac{l}{4} \rceil}\right) \mathrm{nrd}(\beta_{(l-1) \bmod 4+1})}.$$

By the similar process, since each singular value of $\sqrt{B_1}(B_1^{-1} B_3)(\sqrt{B_1})^{-1}$ and $\sqrt{B_1}(B_1^{-1} B_4)(\sqrt{B_1})^{-1}$ is $\frac{1}{\sqrt{p}}$ where $\langle x, y\rangle_j = x^T B_3 y$ and $\langle x, y\rangle_k = x^T B_4 y$ for

$$\langle (a_1, b_1, c_1, d_1), (a_2, b_2, c_2, d_2)\rangle_j := a_1 c_2 + b_1 d_2 + c_1 a_2 + d_1 b_2,$$

$$\langle a_1, b_1, c_1, d_1), (a_2, b_2, c_2, d_2)\rangle_k := a_1 d_2 + b_1 c_2 + c_1 b_2 + d_1 a_2,$$

we have

$$|M_{3,l}| \leq \frac{1}{\sqrt{p}} \sqrt{\mathrm{nrd}\left(\alpha_{\lceil \frac{l}{4} \rceil}\right) \mathrm{nrd}(\beta_{(l-1) \bmod 4+1})},$$

$$|M_{4,l}| \leq \frac{1}{\sqrt{p}} \sqrt{\mathrm{nrd}\left(\alpha_{\lceil \frac{l}{4} \rceil}\right) \mathrm{nrd}(\beta_{(l-1) \bmod 4+1})}.$$

□

A.2 Proof of Lemma 3

There is a $\mathcal{O}_0, \mathcal{O}$-connecting ideal I by Proposition 13.3.4 and Lemma 17.4.6 of [18]. (See Appendix G) Then,

$$\det(\mathcal{O})\mathrm{nrd}(I)^2 = \det(I) = \det(\mathcal{O}_0)\mathrm{nrd}(I)^2$$

by Lemma 15. Clearing each term $\mathrm{nrd}(I)^2$,

$$\det(\mathcal{O}) = \det(\mathcal{O}_0) = \frac{p}{4}$$

by Lemma 16.

□

Lemma 15. *Let I be a left/right-ideal of a maximal order $\mathcal{O}$ in $B_{p,\infty}$. Then,*

$$\det(I) = \det(\mathcal{O})\mathrm{nrd}(I)^2.$$

Proof. By Theorem 16.1.3 of [18], we have

$$[\mathcal{O} : I]_{\mathbb{Z}} = \mathrm{nrd}(I)^2.$$

By Theorem 15.2.15 of [18], we have

$$\mathrm{disc}(I) = [\mathcal{O} : I]^2_{\mathbb{Z}}\mathrm{disc}(\mathcal{O}).$$

By the definition of discriminant [18],

$$\mathrm{disc}(\mathcal{O}) = \det(\mathcal{O})^2, \quad \mathrm{disc}(I) = \det(I)^2.$$

Combining these results, we conclude that

$$\det(I) = \det(\mathcal{O})\mathrm{nrd}(I)^2.$$

□

Lemma 16. $\det(\mathcal{O}_0) = \frac{p}{4}$

Proof. $\mathcal{O}_0 = \langle 1, i, (1+k)/2, (i+j)/2 \rangle$ so its Gram matrix is

$$G = \begin{pmatrix} 1 & 0 & \frac{1}{2} & 0 \\ 0 & 1 & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1+p}{4} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1+p}{4} \end{pmatrix}.$$

Then, $\det G = \det \begin{pmatrix} 1 & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & \frac{1+p}{4} & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1+p}{4} \end{pmatrix} = \det \begin{pmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1+p}{4} \end{pmatrix}^2 = \left(\frac{p}{4}\right)^2.$

Hence, by 1.11 of [31],

$$\det(\mathcal{O}_0) = \sqrt{\det G} = \frac{p}{4}.$$

□

A.3 Proof of Lemma 9

By Proposition 16.4.3 of [18] (see Appendix G), $N(I_i) = \mathrm{nrd}(I_i)^2 = [\mathcal{O} : I_i]$ over $\mathbb{Z}$, for $i = 1, 2$. Given modules A and C, we have a short exact sequence

$$0 \to A \to A \oplus C \to C \to 0$$

by [32]. Applying this result, we have two short exact sequences

$$0 \to (I_1 + I_2)/I_1 \to \mathcal{O}/I_1 \to \mathcal{O}/(I_1 + I_2) \to 0,$$

$$0 \to (I_1 + I_2)/I_2 \to \mathcal{O}/I_2 \to \mathcal{O}/(I_1 + I_2) \to 0,$$

by third isomorphism theorem. Let $q : \mathcal{O} \to \mathcal{O}/(I_1 + I_2)$ be a canonical quotient map and let

$$\alpha_1 : \mathcal{O}/I_1 \to \mathcal{O}/(I_1 + I_2), \quad r \bmod I_1 \mapsto q(r),$$

$$\alpha_2 : \mathcal{O}/I_2 \to \mathcal{O}/(I_1 + I_2), \quad r \bmod I_2 \mapsto q(r).$$

By A.1.9 of [33] (see Appendix G), we have a unique $\mathcal{O}$-linear map $\Phi : \mathcal{O}/I_1 \oplus \mathcal{O}/I_2 \to \mathcal{O}/(I_1 + I_2)$ which is surjective since q is surjective and $\Phi(x, y) = q(x) - q(y),\ \forall (x, y) \in \mathcal{O}/I_1 \oplus \mathcal{O}/I_2$. Then, since $\ker(\Phi) \cong \mathcal{O}/(I_1 \cap I_2)$ by second isomorphism theorem, we have a short exact sequence

$$0 \to \mathcal{O}/(I_1 \cap I_2) \to \mathcal{O}/I_1 \oplus \mathcal{O}/I_2 \to \mathcal{O}/(I_1 + I_2) \to 0$$

by definition.

By Prop.6.9. of [34] (see Appendix G), length l of modules is an additive function. Also, $v_p(\#M) = l(M_p)$ by Lemma 17, where v_p is a p-adic valuation. Hence, combining these results, we have

$$\#\mathcal{O}/(I_1 \cap I_2) \cdot \#\mathcal{O}/(I_1 + I_2) = \#\mathcal{O}/I_1 \cdot \#\mathcal{O}/I_2.$$

Since $[\mathcal{O} : I] := \#\mathcal{O}/I$, we have

$$N(I_1 \cap I_2) = \frac{N(I_1)N(I_2)}{N(I_1 + I_2)}$$

so by the fact that $\mathrm{nrd}(I) = \sqrt{N(I)}$, we can conclude that

$$\mathrm{nrd}(I_1 \cap I_2) \leq \mathrm{nrd}(I_1)\mathrm{nrd}(I_2).$$

□

Lemma 17. *Let M be a finite additive group and M_p be a Sylow p-subgroup of M. Then, $v_p(\#M) = l(M_p)$ for every prime p dividing the $\#M$.*

Proof. There is a p-primary decomposition of M

$$M \cong \bigoplus_p M_p.$$

By Jordan-Holder theorem, every finite group has its unique composition series so the length l of a module can also be defined as the length of its composition series.
Since M_p is a finite p-group,

$$\#M_p = p^{l(M_p)}$$

by definition. Applying the fact that l is an additive function and v_p is a multiplicative function,

$$\#M = \#\bigoplus_p M_p = \prod_p \#M_p = \prod_p p^{l(M_p)}.$$

Then, by the definition of p-adic valuation, we gets

$$v_p(\#M) = l(M_p).$$

□

A.4 Proof of Lemma 10

Let L be the $\mathbb{Z}$-submodule of $\mathcal{O}$ generated by the columns of C. By construction, $L = r_1 I_1 + r_2 I_2$.

Let $H := UC$ be the Hermite normal form of C for some unimodular $U \in \mathrm{GL}_4(\mathbb{Z})$, and let H' be the 4×4 matrix consisting of the nonzero columns of H. As in Appendix A.5, unimodular left multiplication does not change the index, and for a full-rank column lattice we have

$$\Delta \;=\; \gcd\{\det(A) \;:\; A \text{ is a } 4\times 4 \text{ submatrix of } C\} \;=\; [\mathcal{O} : L].$$

Since $r_1 I_1 \subseteq L$ and $r_2 I_2 \subseteq L$, we have

$$[O : L] \;\mid\; [O : r_1 I_1] \quad \text{and} \quad [O : L] \;\mid\; [O : r_2 I_2].$$

Since $\Delta = \prod_{i=1}^{4} H'_{i,i}$, we have $[O : I] = \mathrm{nrd}(I)^2$. Then, it follows that

$$\Delta = [\mathcal{O} : L] \;\mid\; \gcd\big(\mathrm{nrd}(r_1 I_1)^2,\ \mathrm{nrd}(r_2 I_2)^2\big).$$

Finally, since $\mathrm{nrd}(rI) = r^2 \mathrm{nrd}(I)$, $\mathrm{nrd}(rI)^2 = r^4 \mathrm{nrd}(I)^2$, and therefore

$$\Delta \;\le\; \gcd\left(\mathrm{nrd}(r_1 I_1)^2, \mathrm{nrd}(r_2 I_2)^2\right) \;\le\; \min\left(r_1^4\,\mathrm{nrd}(I_1)^2, r_2^4\,\mathrm{nrd}(I_2)^2\right).$$

□

A.5 Proof of Lemma 11

Let $\Delta = \gcd\{\det(A) \mid A \text{ is a } 4 \times 4 \text{ submatrix of } M\}$. We have the Hermite Normal Form of M, $H := UM$ for a unimodular matrix U. Since $\mathcal{O}_1$ is a free $\mathbb{Z}$-module of rank 4, we have

$$\mathcal{O}_1 \cong \mathbb{Z}^4.$$

Let L be a sublattice of $\mathcal{O}_1$ whose basis is the set of column vectors of M. Then,

$$L \cong \{Mx \mid x \in \mathbb{Z}^{16}\} = \{(U^{-1}H)x \mid x \in \mathbb{Z}^{16}\} = \{U^{-1}(Hx) \mid x \in \mathbb{Z}^{16}\} = U^{-1}H\mathbb{Z}^{16}.$$

Since U^{-1} is unimodular, it is an automorphism of $\mathbb{Z}^4$. So it is trivial that it does not change the index. So,

$$[\mathcal{O}_1 : L] = [\mathbb{Z}^4 : U^{-1}H\mathbb{Z}^{16}] = [\mathbb{Z}^4 : H\mathbb{Z}^{16}] = [\mathbb{Z}^4 : H'\mathbb{Z}^4].$$

where H' is a 4×4 matrix whose each column is a nonzero column of H. Since H' is a Hermite Normal Form, by letting $d_i := H'_{i,i}$ for $1 \leq i \leq 4$,

$$[\mathbb{Z}^4 : H'\mathbb{Z}^4] = \prod_{i=1}^{4} d_i.$$

by Thm 2.4.13 of [35]. (See Appendix G)

Since each $r_1 r_2 \alpha_i \beta_j$ is a generator for both L and $r_1 r_2 I_1 I_2$, it is clear that

$$L = r_1 r_2 I_1 I_2.$$

Now, since $\Delta = \prod_{i=1}^{4} d_i$ [23] (see Appendix G), we have

$$\mathrm{nrd}(r_1 I_1)^2 \mathrm{nrd}(r_2 I_2)^2 = \mathrm{nrd}(r_1 r_2 I_1 I_2)^2 = [\mathcal{O}_1 : r_1 r_2 I_1 I_2] = [\mathcal{O}_1 : L] = \Delta$$

□

B Implementation

Based on the uniform bound, we implemented SQIsign in C replacing GMP library with fixed-precision integer arithmetic. We can compute the necessary number of limbs by the uniform bound, where we set each limb size 64-bit. (See Table 3)

We also reflected the modified ideal multiplication in our implementation. Moreover, we modified KeyGen to ensure that our hypothesis about the reduced norm of I_{sk} is valide, by a rejection sampling. We implemented our proof-of-concept source code based on the original Round-2 SQIsign implementation[2]. Our source code is available at:

`https://github.com/munsanwon2/SQIsign-Fixed-Precision`

[2] https://github.com/SQIsign/the-sqisign

Table 3: Number of limbs by the uniform bound.

	NIST-I	**NIST-III**	**NIST-V**
$\lceil \log_2 p \rceil$	251	383	505
Uniform maximum bit	7,026	10,713	14,150
Necessary number of limbs (uint64_t)	110	168	222
Necessary bit size	7,040	10,752	14,208

Algorithm 10 Cornacchia(q, m)

Input: $q, m \in \mathbb{Z}$ with m prime and $1 \leq q \leq m$.
Output: $x, y \in \mathbb{Z}$ such that $x^2 + qy^2 = m$ if such x, y exist, non-existence indicator $\perp$ otherwise.

if $-q$ is not a square modulo m **then**
  return $\perp$
end if
if $m = 2$ **then**
  if $q = 1$ **then**
    return $1, 1$
  else
    return $\perp$
  end if
end if
$s \leftarrow$ **ModularSQRT**$(-q \bmod m,\ m)$ *// Now m is an odd prime and $-q$ square modulo m*
$r \leftarrow m$
while $s^2 > m$ **do**
  $r, s \leftarrow s,\ (r \bmod s)$
end while
$x, y \leftarrow s,\ \sqrt{\frac{m - s^2}{q}}$
if $x^2 + qy^2 = m$ and $y \in \mathbb{Z}$ **then**
  return r, s
else
  return $\perp$
end if

C Proof for some additional subroutines

Now we present two subroutines of the algorithm **RandomIdealGivenNorm**.

The algorithm **Cornacchia** is as followed: (Algorithm 10)

Lemma 18. *In **Cornacchia**, the maximum value of the size of integers is less than m.*

Algorithm 11 GeneralizedRepresentInteger$(M, \omega, \mathcal{O})$

Input: $M \in \mathbb{Z}$ odd such that $M > p$.
Input: $\omega \in B_{p,\infty}$ such that $q = -\omega^2$ is a positive integer and $q \equiv 1 \pmod 4$.
Input: $\mathcal{O}$ a special extremal maximal order containing the suborder $\mathbb{Z}[\omega] + j\,\mathbb{Z}[\omega] \subset \mathcal{O}$, with j from the standard basis of $B_{p,\infty}$.
Output: $\gamma \in \mathcal{O}$ with $\mathrm{nrd}(\gamma)$ equal to M, or raises an exception.

1: Initialize $q \leftarrow -\omega^2$, *counter* $\leftarrow 0$, *bound* $\leftarrow \left\lceil \frac{4M}{p\sqrt{q}} \right\rceil$, and *found* $\leftarrow$ **false**
2: **while** (**not** found) **and** (counter $<$ bound) **do**
3: counter $\leftarrow$ counter $+1$
4: Sample z uniformly from $[1, \ldots, m]$ for $m = \left\lfloor \sqrt{\frac{4M}{p} - q} \right\rfloor$
5: Sample t uniformly from $[-m', \ldots, m']$ for $m' = \left\lfloor \sqrt{\frac{4M - pz^2}{qp}} \right\rfloor$
6: Set $M' \leftarrow 4M - p\,(z^2 + qt^2)$
7: **if** M' is a prime **then**
8: $res \leftarrow$ Cornacchia(q, M')
9: found $\leftarrow (res \neq \perp)$
10: **if** found **then**
11: $x, y \leftarrow res$
12: **if** found **then**
13: $\gamma \leftarrow (x + \omega y + jz + \omega jt)$
14: Set d to be the biggest scalar such that $\gamma/d \in \mathcal{O}$
15: found $\leftarrow (d = 2)$
16: **if** found **then**
17: **return** γ/d
18: **end if**
19: **end if**
20: **end if**
21: **end if**
22: **end while**
23: **raise** Exception("GeneralizedRepresentInteger failed")

Proof. In **line 11**, sizes of integers in **ModularSQRT** depend on the maximum size of integers so it does not grow it.
From **line 11** to **line 16**, $r, s < m$.
In **line 18**, $x \leq \sqrt{m}$ and $y \leq \sqrt{m/q} \leq \sqrt{m}$. □

The algorithm **GeneralizedRepresentInteger** used in **RandomIdealGivenNorm** is as followed: (Algorithm 11)

Lemma 19. *In* ***GeneralizedRepresentInteger****, the maximum value of the size of integers is at most* $\max(M, 4M - pq)$.

Proof. In **line 1**, **line 3**, **line 4**, and **line 5**, the maximum size of integers is less than M.

Algorithm 12 RandomEquivalentPrimeIdeal(I)

Input: I, a left $\mathcal{O}$-ideal.
Output: $J \sim I$ of small prime norm, or raise an exception if unsuccessful.

1: Initialize *counter* $\leftarrow 0$
2: Compute a LLL-reduced basis $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ of I
3: **while** *counter* $< (2 \cdot$ `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 $[-b, \ldots, b]$, for bound $b =$ `QUAT_equiv_bound_coeff`
6: $\beta \leftarrow \sum_{i=1}^{4} c_i \alpha_i$
7: $J \leftarrow \chi_I(\beta)$
8: **if** nrd(J) is prime **then**
9: **return** J
10: **end if**
11: **end while**
12: **raise** Exception("RandomEquivalentPrimeIdeal failed")

In **line 6** we have $|M'| \leq 4M - p$ and,

$$|M'| \leq p(m^2 + qm'^2) - 4M \leq p\left(\frac{4M}{p} - q + q\frac{4M}{qp}\right) - 4M$$
$$= (8M - pq) - 4M = 4M - pq.$$

In **line 8**, since the maximum size of integers in **Cornacchia** is M', it does not increase the maximum size of integers.
In **line 11**, since the size of the outputs of **Cornacchia** is at most $\sqrt{M'}$,

$$|x|, |y| \leq \sqrt{M'} \leq 2\sqrt{M}.$$

In **line 17**, each coefficient of $\gamma/2$ is less than $\sqrt{M}$ and $d > 2$ so the maximum size of integers is $\sqrt{M}$ in this line. □

The subroutine **RandomEquivalentPrimeIdeal** is as followed: (Algorithm 12)

Lemma 20. *In* ***RandomEquivalentPrimeIdeal****, the maximum value of the size of integers during the quaternion operations is less than* 64 *times of the input integers representing the basis of* I*.*

Proof. In **line 2**, computing LLL-reduced basis does not increase the size of integers by Lemma 21.
In **line 3**, *counter* $< (2 \cdot \text{QUAT_equiv_bound_coeff} + 1)^4 = 2^{28}$ [17].
In **line 5**, $|c_1|, |c_2|, |c_3|, |c_4| \leq b = 64$.
In **line 6** and **line 7**, integers representing β are at most 64 times of that of input basis elements of I.

□

Algorithm 13 $\mathrm{L2}_{\eta,\delta}\big((b_0,\ldots,b_{d-1}),\, G\big)$

Input: A basis $(b_0,\ldots,b_{d-1})$ of a d-dimensional quadratic space, the associated $d\times d$ Gram matrix G, (Parameters: $\frac{1}{2}<\eta<1,\quad \frac{1}{4}<\delta<1$).
Output: A (η,δ)-reduced basis of the same lattice, its associated Gram matrix.

1: $\bar{\delta}\leftarrow \mathrm{FLOAT}\big(\frac{\delta+1}{2}\big),\quad \bar{\eta}\leftarrow \mathrm{FLOAT}\big(\frac{\eta+0.5}{2}\big)$
2: $r_{0,0}\leftarrow \mathrm{FLOAT}(G_{0,0}),\quad \mu_{0,0}\leftarrow \mathrm{FLOAT}(1),\quad T\leftarrow [\mathrm{FLOAT}(0),\mathrm{FLOAT}(0),\mathrm{FLOAT}(0),\mathrm{FLOAT}(0)]$
3: $k\leftarrow 1$
4: **while** $k<d$ **do**
5: $\quad (b_0,\ldots,b_{d-1}),\, G,\, r,\, \mu \leftarrow \mathbf{SizeReduce}\big((b_0,\ldots,b_{d-1}),G,k,r,\mu,\bar{\eta}\big)$
6: $\quad T_0\leftarrow \mathrm{FLOAT}\big(G_{k,k}\big)$
7: $\quad$ **for** $i=1$ **to** $k-1$ **do**
8: $\quad\quad T_i\leftarrow T_{i-1}-\mu_{k,i-1}\,r_{k,i-1}$
9: $\quad$ **end for**
10: $\quad s\leftarrow \min\{0\le i\le k \mid T_j<\delta\, r_{j,j} \text{ for all } i\le j<k\}$
11: $\quad$ **if** $k\neq s$ **then**
12: $\quad\quad (b_0,\ldots,b_{d-1}),\, G,\, r,\, \mu \leftarrow \mathbf{InsertBefore}\big((b_0,\ldots,b_{d-1}),G,k,s,r,\mu\big)$
13: $\quad\quad k\leftarrow s$
14: $\quad$ **end if**
15: $\quad k\leftarrow k+1$
16: **end while**
17: **return** $(b_0,\ldots,b_{d-1}),\, G$

The algorithm **L2** used in **RandomEquivalentPrimeIdeal** for computing an LLL-reduced basis is as followed: (Algorithm 13)

Lemma 21. *In the algorithm **L2**, the maximum size of integers does not grow.*

Proof. We know that the algorithm **SizeReduce** does not change the maximum size by Lemma 22.
In **line 8**,

$$T_k = T_0 - \sum_{j=0}^{k-1}\mu_{k,j}r_{k,j} = G_{k,k} - \sum_{j=0}^{k-1}\mu_{k,j}^2 r_{j,j} = r_{k,k} > 0,$$

since

$$\begin{aligned}\|b_k\|^2 &= \left\langle b_k' + \sum_{j=1}^{k-1}\mu_{k,j}b_j',\, b_k' + \sum_{j=1}^{k-1}\mu_{k,j}b_j' \right\rangle \\ &= \langle b_k', b_k'\rangle + 2\sum_{j=1}^{k-1}\mu_{k,j}\langle b_k', b_j'\rangle + \sum_{j=1}^{k-1}\mu_{k,j}^2\langle b_j', b_j'\rangle = G_{k,k} + \sum_{j=1}^{k-1}\mu_{k,j}^2 r_{j,j}\end{aligned}$$

so

$$r_{k,k} = \langle b_k, b'_k \rangle = \left\langle b'_k + \sum_{j=1}^{k-1} \mu_{k,j} b'_j, b'_k \right\rangle = \langle b'_k, b'_k \rangle + \sum_{j=1}^{k-1} \mu_{k,j} \langle b'_j, b'_k \rangle$$

$$= \langle b'_k, b'_k \rangle = \|b_k\|^2 - \sum_{j=0}^{k-1} \mu_{k,j}^2 r_{j,j} = G_{k,k} - \sum_{j=0}^{k-1} \mu_{k,j}^2 r_{j,j}$$

and

$$T_i > T_{i-1}, \ \forall i \ \because \mu_{k,i-1} r_{k,i-1} > 0.$$

In the algorithm **InsertBefore** [17], since its operations are just permuting a basis and its Gram matrix, it does not change the maximum size.
For other lines, it is trivial.

□

It is clear that in the algorithm **ExtendGSOFamily** [17], the maximum size of integers does not grow.

The algorithm **SizeReduce** is as followed: (Algorithm 14)

Lemma 22. *In the algorithm **SizeReduce**, the maximum size of the integers does not grow.*

Proof. Let $\mathcal{G} = \max_{0 \le i,j < d} |G_{i,j}|$.
In **line 8**, since $|\mu_{k,i}| = \left| \frac{r_{k,i}}{r_{i,i}} \right| = \frac{|r_{k,i}|}{|r_{i,i}|} \le \mathcal{G}$ when the lattice is an ideal of $\mathcal{O}_0$, the maximum size of integers does not grow by [18].

In **line 9**, since $\|b_k - X b_i\| \le \|b_k\|$ by [36], the maximum size of integer values does not grow.

In **line 11,14**, since each entry of the Gram matrix is the inner product of two basis vectors, it is at most the maximum length of the size-reduced basis vectors.

In **line 17**, since the updated $\mu_{k,j}$ value is the fraction of inner product values, its size is at most the maximum length of the size-reduced basis vectors.

□

The algorithm **KernelDecomposedToIdeal** is as followed: (Algorithm 15)

Lemma 23. *In **KernelDecomposedToIdeal**, the maximum value of the size of integers during the quaternion operations is less than p.*

Proof. From **line 1** to **line 3**, each vector and matrix is defined over $\mathbb{Z}/2^f\mathbb{Z}$ so its size is less than 2^f.
In **line 4**, since $a, b < 2^f$, generators representing I have integers representing themselves, whose size are less than $2^{f+1} < p$.

□

Algorithm 14 SizeReduce($(b_0, \ldots, b_{d-1})$, G, k, r :: FLOAT, μ :: FLOAT, $\bar{\eta}$ :: FLOAT)

Input: A basis $(b_0, \ldots, b_{d-1})$ of a quadratic space, its $d \times d$ Gram matrix G, index $1 \leq k < d$, the GSO family r, μ up to row $k-1$, parameter $\frac{1}{2}\bar{\eta} < 1$.
Output: $(b_0, \ldots, b_{d-1})$ size-reduced basis of the same lattice, its Gram matrix G, the GSO family r, μ up to row k.

1: done $\leftarrow$ $false$
2: **while** not done **do**
3: $(r, \mu) \leftarrow$**ExtendGSOFamily**(G, k, r, μ)
4: done $\leftarrow$ $true$
5: **for** i **from** $k-1$ **down to** 0 **do**
6: **if** $|\mu_{k,i}| > \bar{\eta}$ **then**
7: done $\leftarrow$ $false$
8: $X \leftarrow \lfloor \mu_{k,i} \rceil$ ▷ Round to the closest integer
9: $b_k \leftarrow b_k - X\, b_i$ ▷ Update basis
10: **for** j **from** 0 **to** $d-1$ **do**
11: $G_{k,j} \leftarrow G_{k,j} - X\, G_{i,j}$ ▷ Update Gram matrix
12: **end for**
13: **for** j **from** 0 **to** $d-1$ **do**
14: $G_{j,k} \leftarrow G_{j,k} - X\, G_{j,i}$ ▷ Update Gram matrix
15: **end for**
16: **for** j **from** 0 **to** $i-1$ **do**
17: $\mu_{k,j} \leftarrow \mu_{k,j} - \mathrm{FLOAT}(X)\, \mu_{i,j}$ ▷ Update μ
18: **end for**
19: **end if**
20: **end for**
21: **end while**
22: **return** $(b_0, \ldots, b_{d-1})$, G, r, μ

The algorithm **RandomEquivalentQuaternion** is as followed: (Algorithm 16)

Lemma 24. *In* ***RandomEquivalentQuaternion****, the maximum value of the size of integers is at most* p^6*.*

Proof. In **line 2**, each entry of Gram matrix has the size at most $4 \cdot \max_{0 \leq i \leq 3} \mathrm{nrd}(b_i)^2$ by Cauchy-Schwartz inequality.
In **line 3**, by Lemma 25, the maximum size of integers is at most

$$D_{\mathrm{rsp}} \cdot D_{\mathrm{mix}}^2 \cdot 2^f < p^6.$$

□

The algorithm **LatticeSampling** is as followed: (Algorithm 17)

Lemma 25. *In* ***LatticeSampling****, the maximum value of the size of integers is at most* B*.*

Algorithm 15 KernelDecomposedToIdeal$_D(c_1, c_2)$

Input: $c_1, c_2 \in \mathbb{Z}$ defining a point $[c_1]P_0 + [c_2]Q_0$ on E_0 of order 2^f generating an isogeny φ.
Output: I_φ, a left $\mathcal{O}_0$-ideal.
1: $[d_1, d_2]^T \leftarrow \mathbf{M}_\theta\,[c_1, c_2]^T$
2: $\mathbf{M} \leftarrow \begin{pmatrix} c_1 & d_1 \\ c_2 & d_2 \end{pmatrix}$
3: $[a, b]^T \leftarrow \mathbf{M}^{-1}\mathbf{M}_i\,[c_1, c_2]^T$
4: $I \leftarrow \mathcal{O}_0\,\langle\, a + b\big(j + \frac{1+k}{2}\big) - i,\ 2^f \,\rangle$
5: **return** I

Algorithm 16 RandomEquivalentQuaternion(L)

Input: A lattice L.
Output: A uniformly sampled $\mathbf{b} \in L$ such that $\mathrm{nrd}(\mathbf{b}) < D_{\mathrm{rsp}} \cdot D_{\mathrm{mix}}^2 \cdot 2^f$.
1: Let (b_0, b_1, b_2, b_3) be a basis of L
2: Compute the Gram matrix $\mathbf{G}$ of (b_0, b_1, b_2, b_3)
3: $\mathbf{b} \leftarrow$ **LatticeSampling**$\big((b_0, b_1, b_2, b_3),\ \mathbf{G},\ D_{\mathrm{rsp}} \cdot D_{\mathrm{mix}}^2 \cdot 2^{f+1}\big)$
4: **return b**

Proof. In **line 2**, by Lemma 21, the maximum size of each entry of U, H is at most 1.
In **line 5**, $B_i \leq \sqrt{B}$.
In **line 8**, since $|x_i| \leq \sqrt{B}$, $|y_i| \leq \sqrt{B}$.
In **line 10**, since the algorithm ensures that $\mathrm{nrd}(b) \leq B$, the maximum value of the size of integers is less than B.

□

The algorithm **ComputeBacktrackingAndNormalize** is as followed: (Algorithm 18)

Lemma 26. *In* ***ComputeBacktrackingAndNormalize****, the maximum value of the size of integers is at most* 2 *times of integers representing* α*.*

Proof. In **line 2** and **line 3**, the maximum size of integers is at most $2 \cdot \max_{0 \leq l \leq 3} |\alpha_l|$, by assuming $\alpha_0, \cdots, \alpha_3$ are integers. (If not, just consider 2α.)
In the other lines, the size does not grow clearly.

□

The algorithm **ComputeEvenNonBacktrackingResponse** is as followed: (Algorithm 19)

Lemma 27. *In* ***ComputeEvenNonBacktrackingResponse*** *using quaternion operations, the maximum size of integers is at most* $\mathrm{nrd}(\alpha)$*.*

Algorithm 17 LatticeSampling($(b_0, \ldots, b_{d-1})$, $\mathbf{G}$, B)

Input: A basis $(b_0, \ldots, b_{d-1})$ of a quadratic space, its $d \times d$ Gram matrix $\mathbf{G}$, an integer $B > 0$.
Output: A uniformly random vector of length $\leq B$ in the lattice generated by $(b_0, \ldots, b_{d-1})$.

Initialize Id to the $d \times d$ identity matrix
$\mathbf{U}, \mathbf{H} \leftarrow \mathbf{L2}_{\eta,\delta}(\mathrm{Id}, \mathbf{G}^{-1})$ *// Parse columns of* $\mathbf{U}$ *and* Id *as basis vectors*
repeat
  for i from 0 up to $d-1$ **do**
    $B_i \leftarrow \lfloor \sqrt{B\,\mathbf{H}_{i,i}} \rfloor$
    Sample x_i uniformly from $[-B_i, B_i]$
  end for
  $(y_0 \ \cdots \ y_{d-1}) \leftarrow (x_0 \ \cdots \ x_{d-1})\,\mathbf{U}^{-1}$
until $(y_0 \ \cdots \ y_{d-1})\,\mathbf{G}\,(y_0 \ \cdots \ y_{d-1})^t \leq B$
return $y_0 b_0 + \cdots + y_{d-1} b_{d-1}$

Algorithm 18 ComputeBacktrackingAndNormalize(α)

Input: A quaternion element α.
Output: A quaternion element α, and the backtracking n.

Write α as $\alpha = \alpha_0 + \alpha_1 i + \alpha_2 j + \alpha_3 k$
$\alpha'_0 \leftarrow \alpha_0 - \alpha_3$ *// $\alpha'_0, \alpha'_1, \alpha'_2, \alpha'_3$ represents α in the $\mathcal{O}_0$-basis*
$\alpha'_1 \leftarrow \alpha_1 - \alpha_2$
$\alpha'_2 \leftarrow \alpha_2/2$
$\alpha'_3 \leftarrow \alpha_3/2$
$g \leftarrow \gcd(\alpha'_0, \alpha'_1, \alpha'_2, \alpha'_3)$
$n \leftarrow \textsc{DyadicValuation}(g)$
$\alpha \leftarrow \alpha/g$
return α, n

Proof. In **line 1**, the maximum size of integers grows at most $\mathrm{nrd}(\alpha)$.
In **line 2**, by Lemma 28, the maximum size of integers is at most $\mathrm{nrd}(I) = 2^r$. □

The algorithm **IdealToKernel** is as followed: (Algorithm 20)

Lemma 28. *In **IdealToKernel**, the maximum size of integers is at most* 2^e.

Proof. In **line 1**, since in the algorithm **IdealGenerator** [17], the size of integers are at most **HNF** of a matrix with entries of size at most the number of the trial for while loop, we can control its size not to exceed 2^e with an acceptable number of trials. So we can say that in **line 1**, the maximum size of integers does not grow.
In **line 2**, since each vector and matrix has its entry in $\mathbb{Z}/2^e\mathbb{Z}$, the maximum size of integers is 2^e.

□

Algorithm 19 ComputeEvenNonBacktrackingResponse(E, P, Q, α, e', r)

Input: The curve E, the points P and Q, the quaternion α and the integers e' and r.
Output: The curve E', the points P' and Q'.

1: $I \leftarrow \mathcal{O}_0\alpha + \mathcal{O}_0(2^r)$
2: $(s, t) \leftarrow$ **IdealToKernel**(I)
3: $K \leftarrow [2^{e'+2}s]P + [2^{e'+2}s]Q$
4: $E', \{P', Q'\} \leftarrow$ **TwoIsogenyChainSmall**$(K, E, r, \{P, Q\}, \mathit{true})$
5: **return** E', P', Q'

Algorithm 20 IdealToKernel(I)

Input: A left $\mathcal{O}_0$-ideal I of norm 2^e for $e \leq f$.
Output: Integers a, b such that $\ker \varphi_I$ is generated by $[a\,2^{f-e}]P_0 + [b\,2^{f-e}]Q_0$.

1: $\alpha \leftarrow$ **IdealGenerator**(I)
2: Compute $[a, b]^T$ in the right kernel of M_α modulo 2^e
3: **return** a, b

D Analysis on Qlapoti

In this section, we analyze the algorithm Qlapoti [28]. The algorithm **Qlapoti** is as follows: (Algorithm 21)

Lemma 29. *In **Qlapoti**, the maximum size of integers is at most*

$$\max\left(2^{16}(\log p)^{16}\mathrm{nrd}(J)^4, \frac{64}{\pi^4}p^2\right).$$

Proof. In **line 1** and **line 2**, by Lemma 30, the maximum size of integers is less than

$$\max\left(2^{16}(\log p)^{16}\mathrm{nrd}(J)^4, \frac{64}{\pi^2}p^2\right).$$

In **line 3** and **line 4**, by Lemma 31, the maximum size of integers is less than

$$\max\left(\frac{8p}{\pi^2}\mathrm{nrd}(J), \mathrm{nrd}(J)^2\right).$$

In **line 8**, the maximum size of integers is less than $\mathrm{nrd}(J)^2$.
In **line 35**, the maximum size of integers is at most $2^e \cdot \mathrm{nrd}(J)$.
For other lines, it is trivial that the maximum size of integers does not grow.

□

In the rest of this section, we prove the lemmas about bounds of subroutines of **Qlapoti**. The algorithm **SmallestEquivIdeal** is as follows: (Algorithm 22)

Algorithm 22 SmallestEquivIdeal(J)

Input: Left $\mathcal{O}_0$-ideal J.
Output: The smallest ideal I equivalent with J.
1: $\alpha \leftarrow$ Shortest vector of J. ▷ See line 1, Algorithm 4
2: **return** $\frac{\bar{\alpha}}{\mathrm{nrd}(J)}J$.

Lemma 30. *In **SmallestEquivIdeal**, the maximum size is less than*

$$\max\left(2^{16}(\log p)^{16}\mathrm{nrd}(J)^4, \frac{64}{\pi^2}p^2\right).$$

Proof. See the proof of Lemma 13, line 1. □

The algorithm **SmallGenerator** is as follows:

Algorithm 23 SmallGenerator(I)

Input: Left $\mathcal{O}_0$-ideal I.
Output: A small generator α of I.
1: $N \leftarrow \mathrm{nrd}(I)$.
2: $(\beta_1, \beta_2, \beta_3, \beta_4) \leftarrow$ **LLL**(I).
3: $c_1, c_2, c_3, c_4 \leftarrow [1, 100000]$.
4: $\alpha \leftarrow \sum_{i=1}^{4} c_i\beta_i$.
5: **if** $\gcd(2a_\alpha r_\alpha, N) = 1$ and $\gcd(\mathrm{nrd}(\alpha), N^2) = N$ **then**
6: **return** α.
7: **else**
8: **goto** line 5.
9: **end if**

Lemma 31. *In **SmallGenerator**, the maximum size is less than*

$$\max\left(\frac{8p}{\pi^2}\mathrm{nrd}(I), \mathrm{nrd}(I)^2\right).$$

Proof. In **line 1**, the maximum size is $\mathrm{nrd}(I)$.
In **line 2**, the maximum size does not grow, similarly with Lemma 21.
In **line 3** and **line 4**, the maximum size is less than $100000\sqrt{\frac{8p}{\pi^2}\mathrm{nrd}(I)} =$

$\frac{200000\sqrt{2}}{\pi}\sqrt{p \cdot \mathrm{nrd}(I)}$ by Lemma 48 of [26].
In **line 5**, the maximum size is

$$\max\left(\frac{8p}{\pi^2}\mathrm{nrd}(I), \mathrm{nrd}(I)^2\right),$$

while computing $\gcd(\mathrm{nrd}(\alpha), N^2)$ by Lemma 48 of [26].

□

Algorithm 21 Qlapoti(J)

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

$I \leftarrow$ **SmallestEquivIdeal**(J)
$n \leftarrow \mathrm{nrd}(I)$
$\alpha \leftarrow$ **SmallGenerator**(I) ▷ write $\alpha = a_\alpha + b_\alpha i + c_\alpha j + d_\alpha k$ for $a_\alpha, b_\alpha, c_\alpha, d_\alpha \in \frac{1}{2}\mathbb{Z}$
$r \leftarrow \mathrm{nrd}(\alpha)/n$
if $\gcd(2a_\alpha, n) \neq 1$ **and** $\gcd(2b_\alpha, n) \neq 1$ **then**
    goto line 3
end if
Compute the shortest (s, t) such that $2a_\alpha s + 2b_\alpha t \equiv 2^e - 2r \pmod{n}$
Set $z := \dfrac{2\big(2^e - 2r - 2a_\alpha s - 2b_\alpha t\big)}{n} - s^2 - t^2$
if $z < 0$ **then**
    goto line 3
end if
if $z \equiv 0 \pmod 4$ **and not** $s = t = 0 \pmod 2$ **then**
    goto line 3
end if
if $z \equiv 1 \pmod 4$ **and not** $s \neq t \pmod 2$ **then**
    goto line 3
end if
if $z \equiv 2 \pmod 4$ **and not** $s = t = 1 \pmod 2$ **then**
    goto line 3
end if
if $z \equiv 3 \pmod 4$ **then**
    goto line 3
end if
sol $\leftarrow$ **Cornacchia**$(1, z)$ ▷ May return $\perp$ if no solution is found.
if sol $= \perp$ **then**
    goto line 3
end if
$z_0, z_1 \leftarrow$ sol ▷ $z_0^2 + z_1^2 = z$
if $z_0 \not\equiv s \pmod 2$ **then**
    swap z_0, z_1
end if
$a_1 \leftarrow (z_0 + s)/2, \;\; b_1 \leftarrow (z_1 + t)/2$
$a_2 \leftarrow s - a_1, \;\; b_2 \leftarrow t - b_1$
$\beta_1 \leftarrow n(a_1 + b_1 i) + \alpha, \;\; \beta_2 \leftarrow n(a_2 + b_2 i) + \alpha$
$I_1 \leftarrow \chi_I(\beta_1), \;\; I_2 \leftarrow \chi_I(\beta_2)$
return I_1, I_2

E Computation of some parameters in SQIsign

We computed some precomputed parameters used in the proof, by the following Python and sagemath codes.

```
print(next_prime(2^512) - 2^512) #D_mix in NIST-I
print(next_prime(2^768) - 2^768) #D_mix in NIST-III
print(next_prime(2^1024) - 2^1024) #D_mix in NIST-V
```

Listing 1.1: Computing Dmix

```

p1 = 5 * 2^248 - 1
p3 = 65 * 2^376 - 1
p5 = 27 * 2^500 - 1

m1 = 2^252 + 65 #QUAT_prime_cofactor in NIST-I
m3 = 2^384 + 369 #QUAT_prime_cofactor in NIST-III
m5 = 2^506 + 51 #QUAT_prime_cofactor in NIST-V

for i in range (1, 10):
    if(i * p1 >= m1):
        print('p1/m1 >= ', i)
        break

for i in range (1, 10):
    if(i * p3 >= m3):
        print('p3/m3 >= ', i)
        break

for i in range (1, 10):
    if(i * p5 >= m5):
        print('p5/m5 >= ', i)
        break
```

Listing 1.2: Computing QUAT_prime_cofactor / p

```
from sage.all import QuaternionAlgebra, QQ, matrix, Integer


def compute_Jt_matrix(p, q_t):
    B = QuaternionAlgebra(QQ, -p, -q_t)

    O0 = B.maximal_order()

    Jt = O0.left_ideal(q_t)

    M = Jt.basis_matrix()
    return matrix(QQ, M)
```

```
13
14
15 def compute_Jt_conj_matrix(p, q_t):
16     M = compute_Jt_matrix(p, q_t)
17
18     D = matrix(QQ, [[1, 0, 0, 0],
19                     [0, -1, 0, 0],
20                     [0, 0, -1, 0],
21                     [0, 0, 0, -1]])
22
23     M_bar = M * D
24     return M_bar
25
26
27 p1 = 5 * 2^248 - 1 #NIST-I
28 p3 = 65 * 2^376 - 1 #NIST-III
29 p5 = 27 * 2^500 - 1 #NIST-V
30
31 q_list1 = [5, 17, 37, 41, 53, 97] #NIST-I
32 q_list3 = [5, 13, 17, 41, 73, 89, 97] #NIST-III
33 q_list5 = [5, 37, 61, 97, 113, 149] #NIST-V
34
35 print('NIST-I Jt')
36 for q in q_list1:
37     M = compute_Jt_matrix(p1, q)
38     M_bar = compute_Jt_conj_matrix(p1, q)
39
40     print(f"Conjugate J_t (\bar{{J_t}}) matrix for q={q}:")
41     print(M_bar)
42
43 print()
44 print('NIST-III Jt')
45 for q in q_list3:
46     M = compute_Jt_matrix(p3, q)
47     M_bar = compute_Jt_conj_matrix(p3, q)
48
49     print(f"Conjugate J_t (\bar{{J_t}}) matrix for q={q}:")
50     print(M_bar)
51
52 print()
53 print('NIST-V Jt')
54 for q in q_list5:
55     M = compute_Jt_matrix(p5, q)
56     M_bar = compute_Jt_conj_matrix(p5, q)
57
58     print(f"Conjugate J_t (\bar{{J_t}}) matrix for q={q}:")
59     print(M_bar)
```

Listing 1.3: Computing J_t

F The Uniform Bound of Naive IdealMultiplication

We denote $\hat{x}$ as the value of x before applying a single line of the algorithm, and $\tilde{x}$ as the value of x before applying the algorithm.

Lemma 32. *While computing the multiplication $I_1 I_2$ of two $\mathcal{O}_0$-ideals I_1, I_2 by computing **HNF** in $\mathbb{Z}$, the maximum size of the integer is as followed:*

When $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}, \{\beta_1, \beta_2, \beta_3, \beta_4\}$ are shortest bases of I_1, I_2, respectively, then for any $1 \le s \le 4$ and $1 \le t \le 16$, we have

$$|M_{s,t}| \le \frac{4000 \cdot (1640^2 + 1)}{p^{25/2}} \frac{2^{395}}{\pi^{158}} \left(\det(\mathcal{O}_0)^{79} \mathrm{nrd}(I_1)^{120} \mathrm{nrd}(I_2)^{38}\right).$$

Proof. For the first loop of $i = 4$, while computing from **line 2** to **line 9**, $|g|, |u|, |v| \le \max(M_{i,i}, M_{i,j})$ since $u \le M_{i,j}/2$ and $v \le M_{i,i}/2$ by [37], so we just need to consider **line 8**.

We use the fact that $|u| \le M_{i,j}/2$ and $|v| \le M_{i,i}/2$.
By applying **line 8** for $j = 3$, we have

$$\begin{aligned}
|M_{k,4}| &\le |u| \cdot |\hat{M}_{k,4}| + |v| \cdot |M_{k,3}| \\
&\le \left(\frac{1}{2} \cdot \frac{4}{\sqrt{p}} \sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_3)}\right) \cdot 4\sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_4)} \\
&\quad + \left(\frac{1}{2} \cdot \frac{4}{\sqrt{p}} \sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_4)}\right) \cdot 4\sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_3)} \\
&= \frac{16}{\sqrt{p}} \mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_3)^{1/2}\mathrm{nrd}(\beta_4)^{1/2}, \text{ for } k = 1, 2
\end{aligned}$$

$$\begin{aligned}
|M_{3,4}| &\le |u| \cdot |\hat{M}_{3,4}| + |v| \cdot |M_{3,3}| \\
&\le \left(\frac{1}{2} \cdot \frac{4}{\sqrt{p}} \sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_3)}\right) \cdot \frac{4}{\sqrt{p}} \sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_4)} \\
&\quad + \left(\frac{1}{2} \cdot \frac{4}{\sqrt{p}} \sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_4)}\right) \cdot \frac{4}{\sqrt{p}} \sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_3)} \\
&= \frac{16}{p} \mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_3)^{1/2}\mathrm{nrd}(\beta_4)^{1/2}
\end{aligned}$$

and the size of $|M_{4,4}|$ does not grow since $M_{4,4} = g$.

By applying **line 8** for $j = 2$, we have

$$\begin{aligned}
|M_{k,4}| &\le |u| \cdot |\hat{M}_{k,4}| + |v| \cdot |M_{k,2}| \\
&\le \left(\frac{1}{2} \cdot \frac{4}{\sqrt{p}} \sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_2)}\right) \cdot \frac{16}{\sqrt{p}} \mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_3)^{1/2}\mathrm{nrd}(\beta_4)^{1/2} \\
&\quad + \left(\frac{1}{2} \cdot \frac{4}{\sqrt{p}} \sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_4)}\right) \cdot 4\sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_2)} \\
&\le \frac{8(\sqrt{p}+4)}{p} \mathrm{nrd}(\alpha_1)^{3/2}\mathrm{nrd}(\beta_2)^{1/2}\mathrm{nrd}(\beta_3)^{1/2}\mathrm{nrd}(\beta_4)^{1/2}, \text{ for } k = 1, 2
\end{aligned}$$

$$\begin{aligned}
|M_{3,4}| &\le |u| \cdot |\hat{M}_{3,4}| + |v| \cdot |M_{3,2}| \\
&\le \left(\frac{1}{2} \cdot \frac{4}{p} \sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_2)}\right) \cdot \frac{16}{p} \mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_3)^{1/2}\mathrm{nrd}(\beta_4)^{1/2} \\
&\quad + \left(\frac{1}{2} \cdot \frac{4}{\sqrt{p}} \sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_4)}\right) \cdot \frac{4}{\sqrt{p}} \sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_2)} \\
&\le \frac{8(p+4)}{p^2} \mathrm{nrd}(\alpha_1)^{3/2}\mathrm{nrd}(\beta_2)^{1/2}\mathrm{nrd}(\beta_3)^{1/2}\mathrm{nrd}(\beta_4)^{1/2}
\end{aligned}$$

and the size of $|M_{4,4}|$ does not grow since $M_{4,4} = g$.

By applying **line 8** for $j = 1$, we have

$$\begin{aligned}
|M_{k,4}| &\le |u| \cdot |\hat{M}_{k,4}| + |v| \cdot |M_{k,1}| \\
&\le \left(\frac{1}{2} \cdot \frac{4}{\sqrt{p}} \sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_1)}\right) \cdot \frac{8(\sqrt{p}+4)}{p} \mathrm{nrd}(\alpha_1)^{3/2}\mathrm{nrd}(\beta_2)^{1/2}\mathrm{nrd}(\beta_3)^{1/2}\mathrm{nrd}(\beta_4)^{1/2} \\
&\quad + \left(\frac{1}{2} \cdot \frac{4}{\sqrt{p}} \sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_4)}\right) \cdot 4\sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_1)} \\
&\le \frac{8(p+2\sqrt{p}+8)}{p^{3/2}} \mathrm{nrd}(\alpha_1)^2 \prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{1/2} \text{ for } k = 1, 2
\end{aligned}$$

$$\begin{aligned}
|M_{3,4}| &\le |u| \cdot |\hat{M}_{k,4}| + |v| \cdot |M_{k,1}| \\
&\le \left(\frac{1}{2} \cdot \frac{4}{\sqrt{p}} \sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_1)}\right) \cdot \frac{8(p+4)}{p^2} \mathrm{nrd}(\alpha_1)^{3/2}\mathrm{nrd}(\beta_2)^{1/2}\mathrm{nrd}(\beta_3)^{1/2}\mathrm{nrd}(\beta_4)^{1/2} \\
&\quad + \left(\frac{1}{2} \cdot \frac{4}{\sqrt{p}} \sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_4)}\right) \cdot \frac{4}{\sqrt{p}} \sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_1)} \\
&\le \frac{8(p^{3/2}+2p+8)}{p^{5/2}} \mathrm{nrd}(\alpha_1)^2 \prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{1/2}
\end{aligned}$$

and the size of $|M_{4,4}|$ does not grow since $M_{4,4} = g$.

For the loop from **line 10** to **line 13**, we can see that **line 11** does not increase the maximum size of integers, so we just need to consider **line 12**.

By applying **line 12** for $j = 1, 2, 3$, we have

$$
\begin{aligned}
|M_{k,j}| = |\hat{M}_{k,j} - gM_{k,4}| &\leq |\hat{M}_{k,j}| + |M_{4,j}| \cdot |M_{k,4}| \\
&\leq 4\sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_j)} + \frac{4}{\sqrt{p}}\sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_j)} \cdot \frac{8(p + 2\sqrt{p} + 8)}{p^{3/2}}\mathrm{nrd}(\alpha_1)^2 \prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{1/2} \\
&\leq \frac{4}{p}\sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_j)}\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{1/2} \\
&\quad + \frac{4}{\sqrt{p}}\sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_j)} \cdot \frac{8(p + 2\sqrt{p} + 8)}{p^{3/2}}\mathrm{nrd}(\alpha_1)^2 \prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{1/2} \\
&\leq \frac{4(9p + 16\sqrt{p} + 64)}{p^2}\mathrm{nrd}(\alpha_1)^{5/2}\mathrm{nrd}(\beta_j)^{1/2}\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{1/2} \\
&\quad \text{for } k = 1, 2
\end{aligned}
$$

since $\dim(I_2) = 4$ so at least two out of $(\beta_1, \beta_2, \beta_3, \beta_4)$ satisfies $c_{\beta_l} d_{\beta_l} \neq 0$, which means that $\mathrm{nrd}(\beta_l) \geq p$.

$$
\begin{aligned}
|M_{3,j}| = |\hat{M}_{3,j} - gM_{3,4}| &\leq |\hat{M}_{3,j}| + |M_{4,j}| \cdot |M_{3,4}| \\
&\leq \frac{4}{\sqrt{p}}\sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_j)} + \frac{4}{\sqrt{p}}\sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_j)} \cdot \frac{8(p^{3/2} + 2p + 8)}{p^{5/2}}\mathrm{nrd}(\alpha_1)^2 \prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{1/2} \\
&\leq \frac{4}{p^{3/2}}\sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_j)}\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{1/2} \\
&\quad + \frac{4}{\sqrt{p}}\sqrt{\mathrm{nrd}(\alpha_1)\mathrm{nrd}(\beta_j)} \cdot \frac{8(p^{3/2} + 2p + 8)}{p^{5/2}}\mathrm{nrd}(\alpha_1)^2 \prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{1/2} \\
&\leq \frac{4(9p^{3/2} + 16p + 64)}{p^3}\mathrm{nrd}(\alpha_1)^{5/2}\mathrm{nrd}(\beta_j)^{1/2}\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{1/2}
\end{aligned}
$$

and $M_{4,j} = 0$ so its size does not grow trivially.

For the loop from **line 14** to **line 18**, we can see that **line 15** and **line 16**, the integer values do not increase the maximum size of integers clearly and no entry of the matrix changes, we just need to consider **line 17**.
We will use the fact that $|r| < |M_{i,i}|,\ |g| \leq |M_{i,j}|$.

By applying **line 17** from $j = 5$ to $j = 16$, we have

$$
\begin{aligned}
|M_{k,j}| = |\hat{M}_{k,j} - gM_{k,4}| &\leq |\hat{M}_{k,j}| + |M_{4,j}| \cdot |M_{k,4}| \\
&\leq 4\sqrt{\mathrm{nrd}\left(\alpha_{\lceil \frac{j}{4} \rceil}\right) \mathrm{nrd}(\beta_{(j-1) \bmod 4+1})} \\
&\quad + \frac{4}{\sqrt{p}}\sqrt{\mathrm{nrd}\left(\alpha_{\lceil \frac{j}{4} \rceil}\right) \mathrm{nrd}(\beta_{(j-1) \bmod 4+1})} \cdot \frac{8(p + 2\sqrt{p} + 8)}{p^{3/2}} \mathrm{nrd}(\alpha_1)^2 \prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{1/2} \\
&\leq \frac{4}{p}\sqrt{\mathrm{nrd}\left(\alpha_{\lceil \frac{j}{4} \rceil}\right) \mathrm{nrd}(\beta_{(j-1) \bmod 4+1})} \prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{1/2} \\
&\quad + \frac{4}{\sqrt{p}}\sqrt{\mathrm{nrd}\left(\alpha_{\lceil \frac{j}{4} \rceil}\right) \mathrm{nrd}(\beta_{(j-1) \bmod 4+1})} \cdot \frac{8(p + 2\sqrt{p} + 8)}{p^{3/2}} \mathrm{nrd}(\alpha_1)^2 \prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{1/2} \\
&\leq \frac{4(9p + 16\sqrt{p} + 64)}{p^2} \mathrm{nrd}(\alpha_1)^2 \mathrm{nrd}\left(\alpha_{\lceil \frac{j}{4} \rceil}\right)^{1/2} \mathrm{nrd}(\beta_{(j-1) \bmod 4+1})^{1/2} \prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{1/2} \\
&\quad \text{for } k = 1, 2
\end{aligned}
$$

$$
\begin{aligned}
|M_{3,j}| = |\hat{M}_{3,j} - gM_{3,4}| &\leq |\hat{M}_{3,j}| + |M_{4,j}| \cdot |M_{3,4}| \\
&\leq \frac{4}{\sqrt{p}}\sqrt{\mathrm{nrd}\left(\alpha_{\lceil \frac{j}{4} \rceil}\right) \mathrm{nrd}(\beta_{(j-1) \bmod 4+1})} \\
&\quad + \frac{4}{\sqrt{p}}\sqrt{\mathrm{nrd}\left(\alpha_{\lceil \frac{j}{4} \rceil}\right) \mathrm{nrd}(\beta_{(j-1) \bmod 4+1})} \cdot \frac{8(p^{3/2} + 2p + 8)}{p^{5/2}} \mathrm{nrd}(\alpha_1)^2 \prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{1/2} \\
&\leq \frac{4}{p^{3/2}}\sqrt{\mathrm{nrd}\left(\alpha_{\lceil \frac{j}{4} \rceil}\right) \mathrm{nrd}(\beta_{(j-1) \bmod 4+1})} \prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{1/2} \\
&\quad + \frac{4}{\sqrt{p}}\sqrt{\mathrm{nrd}\left(\alpha_{\lceil \frac{j}{4} \rceil}\right) \mathrm{nrd}(\beta_{(j-1) \bmod 4+1})} \cdot \frac{8(p^{3/2} + 2p + 8)}{p^{5/2}} \mathrm{nrd}(\alpha_1)^2 \prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{1/2} \\
&\leq \frac{4(9p^{3/2} + 16p + 64)}{p^3} \mathrm{nrd}(\alpha_1)^2 \mathrm{nrd}\left(\alpha_{\lceil \frac{j}{4} \rceil}\right)^{1/2} \mathrm{nrd}(\beta_{(j-1) \bmod 4+1}) \prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{1/2}
\end{aligned}
$$

and the size of $|M_{4,j}|$ does not grow since $M_{4,j} = r$.

From the previous results, we now compute the loop of $i = 3$.

By applying **line 8** for $j = 2$, we have

$$
\begin{aligned}
|M_{k,3}| &\le |u| \cdot |\hat{M}_{k,3}| + |v| \cdot |M_{k,2}| \\
&\le \frac{4(9p^{3/2} + 16p + 64)}{2p^3} \mathrm{nrd}(\alpha_1)^{5/2} \mathrm{nrd}(\beta_2)^{1/2} \prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{1/2} \\
&\quad \times \frac{4(9p + 16\sqrt{p} + 64)}{p^2} \mathrm{nrd}(\alpha_1)^{5/2} \mathrm{nrd}(\beta_3)^{1/2} \prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{1/2} \\
&\quad + \frac{4(9p^{3/2} + 16p + 64)}{2p^3} \mathrm{nrd}(\alpha_1)^{5/2} \mathrm{nrd}(\beta_3)^{1/2} \prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{1/2} \\
&\quad \times \frac{4(9p + 16\sqrt{p} + 64)}{p^2} \mathrm{nrd}(\alpha_1)^{5/2} \mathrm{nrd}(\beta_2)^{1/2} \prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{1/2} \\
&= \frac{16(9p + 16\sqrt{p} + 64)(9p^{3/2} + 16p + 64)}{p^5} \mathrm{nrd}(\alpha_1)^{5} \mathrm{nrd}(\beta_2)^{1/2} \mathrm{nrd}(\beta_3)^{1/2} \prod_{l=1}^{4} \mathrm{nrd}(\beta_l) \\
&\quad \text{for } k = 1, 2
\end{aligned}
$$

and the size of $|M_{3,3}|$ does not grow since $M_{3,3} = g$.
$M_{4,3} = u\hat{M}_{4,3} + vM_{4,2} = u \cdot 0 + v \cdot 0 = 0$ so its size does not grow trivially.

By applying **line 8** for $j = 1$, we have

$$
\begin{aligned}
|M_{k,3}| &\le |u| \cdot |\hat{M}_{k,3}| + |v| \cdot |M_{k,1}| \\
&\le \frac{4(9p^{3/2} + 16p + 64)}{2p^3} \mathrm{nrd}(\alpha_1)^{5/2} \mathrm{nrd}(\beta_2)^{1/2} \prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{1/2} \\
&\quad \times \frac{16(9p + 16\sqrt{p} + 64)(9p^{3/2} + 16p + 64)}{p^5} \mathrm{nrd}(\alpha_1)^{5} \mathrm{nrd}(\beta_2)^{1/2} \mathrm{nrd}(\beta_3)^{1/2} \prod_{l=1}^{4} \mathrm{nrd}(\beta_l) \\
&\quad + \frac{4(9p^{3/2} + 16p + 64)}{2p^3} \mathrm{nrd}(\alpha_1)^{5/2} \mathrm{nrd}(\beta_3)^{1/2} \prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{1/2} \\
&\quad \times \frac{4(9p + 16\sqrt{p} + 64)}{p^2} \mathrm{nrd}(\alpha_1)^{5/2} \mathrm{nrd}(\beta_1)^{1/2} \prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{1/2} \\
&\le \frac{32(9p + 16\sqrt{p} + 64)(9p^{3/2} + 16p + 64) \cdot p^2}{p^8} \mathrm{nrd}(\alpha_1)^{15/2} \mathrm{nrd}(\beta_2) \mathrm{nrd}(\beta_3)^{1/2} \prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{3/2} \\
&\quad + \frac{8(9p + 16\sqrt{p} + 64)(9p^{3/2} + 16p + 64)}{p^5 \cdot p} \mathrm{nrd}(\alpha_1)^{5} \mathrm{nrd}(\beta_1)^{1/2} \mathrm{nrd}(\beta_3)^{1/2} \prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{3/2} \\
&\le \frac{40(9p + 16\sqrt{p} + 64)(9p^{3/2} + 16p + 64)}{p^6} \mathrm{nrd}(\alpha_1)^{15/2} \mathrm{nrd}(\beta_2)^{1/2} \frac{\prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^2}{\mathrm{nrd}(\beta_4)^{1/2}}, \text{ for } k = 1, 2
\end{aligned}
$$

since $p^2 \ge 9p^{3/2} + 16p + 64$ in SQIsign.

The size of $|M_{3,3}|$ does not grow since $M_{3,3} = g$.
$M_{4,3} = u\hat{M}_{4,3} + vM_{4,2} = u \cdot 0 + v \cdot 0 = 0$ so its size does not grow trivially.

By applying **line 12** for $j = 1, 2$, we have

$$\begin{aligned}
|M_{k,j}| = |\hat{M}_{k,j} - gM_{k,3}| &\le |\hat{M}_{k,j}| + |M_{3,j}| \cdot |M_{k,3}| \\
&\le \frac{4(9p + 16\sqrt{p} + 64)}{p^2}\mathrm{nrd}(\alpha_1)^{5/2}\mathrm{nrd}(\beta_j)^{1/2}\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{1/2} \\
&\quad + \frac{4(9p^{3/2} + 16p + 64)}{p^3}\mathrm{nrd}(\alpha_1)^{5/2}\mathrm{nrd}(\beta_j)^{1/2}\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{1/2} \\
&\quad \times \frac{40(9p + 16\sqrt{p} + 64)(9p^{3/2} + 16p + 64)}{p^6}\mathrm{nrd}(\alpha_1)^{15/2}\mathrm{nrd}(\beta_2)^{1/2}\frac{\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^2}{\mathrm{nrd}(\beta_4)^{1/2}} \\
&\le \frac{4(9p + 16\sqrt{p} + 64)}{p^2 \cdot p^{7/2}}\mathrm{nrd}(\alpha_1)^{5/2}\mathrm{nrd}(\beta_j)^{1/2}\frac{\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{5/2}}{\mathrm{nrd}(\beta_4)^{1/2}} \\
&\quad + \frac{160(9p + 16\sqrt{p} + 64) \cdot p^{7/2}}{p^9}\mathrm{nrd}(\alpha_1)^{10}\mathrm{nrd}(\beta_j)^{1/2}\mathrm{nrd}(\beta_2)^{1/2}\frac{\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{5/2}}{\mathrm{nrd}(\beta_4)^{1/2}} \\
&\le \frac{164(9p + 16\sqrt{p} + 64)}{p^{11/2}}\mathrm{nrd}(\alpha_1)^{10}\mathrm{nrd}(\beta_j)^{1/2}\mathrm{nrd}(\beta_2)^{1/2}\frac{\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{5/2}}{\mathrm{nrd}(\beta_4)^{1/2}}, \text{ for } k = 1, 2
\end{aligned}$$

since $p^{7/2} > (9p^{3/2} + 16p + 64)^2$ in SQIsign.

$M_{3,j} = 0$ so its size does not grow trivially.
$M_{4,j} = \hat{M}_{4,j} - gM_{4,3} = 0 - g \cdot 0 = 0$ so its size does not grow trivially.

By applying **line 17** for $j = 4$, we have

$$\begin{aligned}
|M_{k,4}| = |\hat{M}_{k,4} - gM_{k,3}| &\le |\hat{M}_{k,4}| + |M_{3,4}| \cdot |M_{k,3}| \\
&\le \frac{8(p + 2\sqrt{p} + 8)}{p^{3/2}}\mathrm{nrd}(\alpha_1)^2\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{1/2} \\
&\quad + \frac{8(p^{3/2} + 2p + 8)}{p^{5/2}}\mathrm{nrd}(\alpha_1)^2\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{1/2} \\
&\quad \times \frac{40(9p + 16\sqrt{p} + 64)(9p^{3/2} + 16p + 64)}{p^6}\mathrm{nrd}(\alpha_1)^{15/2}\mathrm{nrd}(\beta_2)^{1/2}\frac{\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^2}{\mathrm{nrd}(\beta_4)^{1/2}} \\
&\le \frac{8(p + 2\sqrt{p} + 8)}{p^{3/2} \cdot p^{7/2}}\mathrm{nrd}(\alpha_1)^2\frac{\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{5/2}}{\mathrm{nrd}(\beta_4)^{1/2}} \\
&\quad + \frac{320(9p + 16\sqrt{p} + 64) \cdot p^{7/2}}{p^{17/2}}\mathrm{nrd}(\alpha_1)^{19/2}\mathrm{nrd}(\beta_2)^{1/2}\frac{\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{5/2}}{\mathrm{nrd}(\beta_4)^{1/2}} \\
&\le \frac{8(361p + 642\sqrt{p} + 2568)}{p^5}\mathrm{nrd}(\alpha_1)^{19/2}\mathrm{nrd}(\beta_2)^{1/2}\frac{\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{5/2}}{\mathrm{nrd}(\beta_4)^{1/2}}, \text{ for } k = 1, 2
\end{aligned}$$

since $p^{7/2} > (p^{3/2} + 2p + 8)(9p^{3/2} + 16p + 64)$ in SQIsign.

The size of $|M_{3,4}|$ does not grow since $|M_{3,4}| = r$.
$M_{4,4} = \hat{M}_{4,4} - gM_{4,3} = \hat{M}_{4,4} - g \cdot 0 = \hat{M}_{4,4}$ so the size of $M_{4,4}$ does not grow.

By applying **line 17** from $j = 5$ to $j = 16$, we have

$$
\begin{aligned}
|M_{k,j}| &= |\hat{M}_{k,j} - gM_{k,3}| \le |\hat{M}_{k,j}| + |M_{3,j}| \cdot |M_{k,3}| \\
&\le \frac{4(9p + 16\sqrt{p} + 64)}{p^2} \mathrm{nrd}(\alpha_1)^2 \mathrm{nrd}\left(\alpha_{\lceil \frac{j}{4} \rceil}\right)^{1/2} \mathrm{nrd}(\beta_{((j-1) \bmod 4)+1})^{1/2} \prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{1/2} \\
&\quad + \frac{4(9p^{3/2} + 16p + 64)}{p^3} \mathrm{nrd}(\alpha_1)^2 \mathrm{nrd}\left(\alpha_{\lceil \frac{j}{4} \rceil}\right)^{1/2} \mathrm{nrd}(\beta_3) \prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{1/2} \\
&\quad \times \frac{40(9p + 16\sqrt{p} + 64)(9p^{3/2} + 16p + 64)}{p^6} \mathrm{nrd}(\alpha_1)^{15/2} \mathrm{nrd}(\beta_2)^{1/2} \frac{\prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^2}{\mathrm{nrd}(\beta_4)^{1/2}} \\
&\le \frac{4(9p + 16\sqrt{p} + 64)}{p^2 \cdot p^{7/2}} \mathrm{nrd}(\alpha_1)^2 \mathrm{nrd}\left(\alpha_{\lceil \frac{j}{4} \rceil}\right)^{1/2} \mathrm{nrd}(\beta_{((j-1) \bmod 4)+1})^{1/2} \prod_{l=1}^{4} \frac{\mathrm{nrd}(\beta_l)^{5/2}}{\mathrm{nrd}(\beta_4)^{1/2}} \\
&\quad + \frac{160(9p + 16\sqrt{p} + 64) \cdot p^{7/2}}{p^9} \mathrm{nrd}(\alpha_1)^{19/2} \mathrm{nrd}\left(\alpha_{\lceil \frac{j}{4} \rceil}\right)^{1/2} \mathrm{nrd}(\beta_{((j-1) \bmod 4)+1})^{1/2} \mathrm{nrd}(\beta_2)^{1/2} \frac{\prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{5/2}}{\mathrm{nrd}(\beta_4)^{1/2}} \\
&\le \frac{640(9p + 16\sqrt{p} + 64)}{p^{11/2}} \mathrm{nrd}(\alpha_1)^{19/2} \mathrm{nrd}\left(\alpha_{\lceil \frac{j}{4} \rceil}\right)^{1/2} \mathrm{nrd}(\beta_{((j-1) \bmod 4)+1})^{1/2} \mathrm{nrd}(\beta_2)^{1/2} \frac{\prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{5/2}}{\mathrm{nrd}(\beta_4)^{1/2}} \\
&\quad \text{for } k = 1, 2
\end{aligned}
$$

since $p^{7/2} > (9p^{3/2} + 16p + 64)^2$ in SQIsign.

and the size of $|M_{3,j}|$ does not grow since $M_{3,j} = r$.
$M_{4,j} = \hat{M}_{4,j} - gM_{4,3} = \hat{M}_{4,j} - g \cdot 0 = \hat{M}_{4,j}$ so the size of $|M_{4,j}|$ does not grow.

From the previous results, we now compute the loop of $i = 2$.

By applying **line 8** for $j = 1$, we have

$$
\begin{aligned}
|M_{1,2}| &= |u\hat{M}_{1,2} + vM_{1,1}| \le |u| \cdot |\hat{M}_{1,2}| + |v| \cdot |M_{1,1}| \\
&\le \frac{164(9p + 16\sqrt{p} + 64)}{2p^{11/2}} \mathrm{nrd}(\alpha_1)^{10} \mathrm{nrd}(\beta_1)^{1/2} \mathrm{nrd}(\beta_2)^{1/2} \frac{\prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{5/2}}{\mathrm{nrd}(\beta_4)^{1/2}} \\
&\quad \times \frac{164(9p + 16\sqrt{p} + 64)}{p^{11/2}} \mathrm{nrd}(\alpha_1)^{10} \mathrm{nrd}(\beta_2) \frac{\prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{5/2}}{\mathrm{nrd}(\beta_4)^{1/2}} \\
&\quad + \frac{164(9p + 16\sqrt{p} + 64)}{2p^{11/2}} \mathrm{nrd}(\alpha_1)^{10} \mathrm{nrd}(\beta_2) \frac{\prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{5/2}}{\mathrm{nrd}(\beta_4)^{1/2}} \\
&\quad \times \frac{164(9p + 16\sqrt{p} + 64)}{p^{11/2}} \mathrm{nrd}(\alpha_1)^{10} \mathrm{nrd}(\beta_1)^{1/2} \mathrm{nrd}(\beta_2)^{1/2} \frac{\prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{5/2}}{\mathrm{nrd}(\beta_4)^{1/2}} \\
&= \frac{164^2(9p + 16\sqrt{p} + 64)^2}{p^{11}} \mathrm{nrd}(\alpha_1)^{20} \mathrm{nrd}(\beta_1)^{1/2} \mathrm{nrd}(\beta_2)^{3/2} \frac{\prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^5}{\mathrm{nrd}(\beta_4)}
\end{aligned}
$$

and the size of $|M_{2,2}|$ does not grow since $M_{2,2} = g$.
$M_{3,2} = u\hat{M}_{3,2} + vM_{3,1} = u \cdot 0 + v \cdot 0 = 0$ so its size does not grow trivially.
$M_{4,2} = u\hat{M}_{4,2} + vM_{4,1} = u \cdot 0 + v \cdot 0 = 0$ so its size does not grow trivially.

By applying **line 12** for $j = 1$, we have

$$\begin{aligned}
|M_{1,1}| = |\hat{M}_{1,1} - gM_{1,2}| &\le |\hat{M}_{1,1}| + |g| \cdot |M_{1,2}| \\
&\le \frac{164(9p + 16\sqrt{p} + 64)}{p^{11/2}} \mathrm{nrd}(\alpha_1)^{10} \mathrm{nrd}(\beta_1)^{1/2} \mathrm{nrd}(\beta_2)^{1/2} \frac{\prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{5/2}}{\mathrm{nrd}(\beta_4)^{1/2}} \\
&\quad + \frac{164(9p + 16\sqrt{p} + 64)}{p^{11/2}} \mathrm{nrd}(\alpha_1)^{10} \mathrm{nrd}(\beta_1)^{1/2} \mathrm{nrd}(\beta_2)^{1/2} \frac{\prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{5/2}}{\mathrm{nrd}(\beta_4)^{1/2}} \\
&\quad \times \frac{164^2(9p + 16\sqrt{p} + 64)^2}{p^{11}} \mathrm{nrd}(\alpha_1)^{20} \mathrm{nrd}(\beta_1)^{1/2} \mathrm{nrd}(\beta_2)^{3/2} \frac{\prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{5}}{\mathrm{nrd}(\beta_4)} \\
&\le \frac{164(9p + 16\sqrt{p} + 64)}{p^{11/2} \cdot p^9} \mathrm{nrd}(\alpha_1)^{10} \mathrm{nrd}(\beta_1)^{1/2} \mathrm{nrd}(\beta_2)^{1/2} \frac{\prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{15/2}}{\mathrm{nrd}(\beta_4)^{3/2}} \\
&\quad + \frac{164^3(9p + 16\sqrt{p} + 64)(10p)^2}{p^{33/2}} \mathrm{nrd}(\alpha_1)^{30} \mathrm{nrd}(\beta_1) \mathrm{nrd}(\beta_2)^2 \frac{\prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{15/2}}{\mathrm{nrd}(\beta_4)^{3/2}} \\
&= \frac{164 \cdot (1640^2 + 1)(9p + 16\sqrt{p} + 64)}{p^{29/2}} \mathrm{nrd}(\alpha_1)^{30} \mathrm{nrd}(\beta_1) \mathrm{nrd}(\beta_2)^2 \frac{\prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{15/2}}{\mathrm{nrd}(\beta_4)^{3/2}}
\end{aligned}$$

since $10p > 9p + 16\sqrt{p} + 64$ in SQIsign.

$M_{2,1} = 0$ so its size does not grow trivially.
$M_{3,1} = \hat{M}_{3,1} - gM_{3,2} = 0 - g \cdot 0 = 0$ so its size does not grow trivially.
$M_{4,1} = \hat{M}_{4,1} - gM_{4,2} = 0 - g \cdot 0 = 0$ so its size does not grow trivially.

By applying **line 17** for $j = 3$, we have

$$\begin{aligned}
|M_{1,3}| = |\hat{M}_{1,3} - gM_{1,2}| &\le |\hat{M}_{1,3}| + |g| \cdot |M_{1,2}| \\
&\le \frac{40(9p + 16\sqrt{p} + 64)(9p^{3/2} + 16p + 64)}{p^6} \mathrm{nrd}(\alpha_1)^{15/2} \mathrm{nrd}(\beta_2)^{1/2} \frac{\prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{2}}{\mathrm{nrd}(\beta_4)^{1/2}} \\
&\quad + \frac{40(9p + 16\sqrt{p} + 64)(9p^{3/2} + 16p + 64)}{p^6} \mathrm{nrd}(\alpha_1)^{15/2} \mathrm{nrd}(\beta_2)^{1/2} \frac{\prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{2}}{\mathrm{nrd}(\beta_4)^{1/2}} \\
&\quad \times \frac{164^2(9p + 16\sqrt{p} + 64)^2}{p^{11}} \mathrm{nrd}(\alpha_1)^{20} \mathrm{nrd}(\beta_1)^{1/2} \mathrm{nrd}(\beta_2)^{3/2} \frac{\prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{5}}{\mathrm{nrd}(\beta_4)} \\
&\le \frac{40(9p + 16\sqrt{p} + 64)(9p^{3/2} + 16p + 64)}{p^6 \cdot p^9} \mathrm{nrd}(\alpha_1)^{15/2} \mathrm{nrd}(\beta_2)^{1/2} \frac{\prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{7}}{\mathrm{nrd}(\beta_4)^{3/2}} \\
&\quad + \frac{40 \cdot 164^2(9p + 16\sqrt{p} + 64)(9p^{3/2} + 16p + 64) \cdot (10p)^2}{p^{17}} \mathrm{nrd}(\alpha_1)^{55/2} \mathrm{nrd}(\beta_1)^{1/2} \mathrm{nrd}(\beta_2)^2 \frac{\prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{7}}{\mathrm{nrd}(\beta_4)^{3/2}} \\
&\le \frac{40 \cdot (1640^2 + 1)(9p + 16\sqrt{p} + 64)(9p^{3/2} + 16p + 64)}{p^{15}} \mathrm{nrd}(\alpha_1)^{55/2} \mathrm{nrd}(\beta_1)^{1/2} \mathrm{nrd}(\beta_2)^2 \frac{\prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^{7}}{\mathrm{nrd}(\beta_4)^{3/2}}
\end{aligned}$$

since $10p > 9p + 16\sqrt{p} + 64$ in SQIsign.

$M_{2,3} = \hat{M}_{2,3} - gM_{2,2} = \hat{M}_{2,3} - (\hat{M}_{2,3} - r) = r$ so the size of $M_{2,3}$ does not grow.
$M_{3,3} = \hat{M}_{3,3} - gM_{3,2} = \hat{M}_{3,3} - g \cdot 0 = \hat{M}_{3,3}$ so its size does not grow.
$M_{4,3} = \hat{M}_{4,3} - gM_{4,2} = \hat{M}_{4,3} - g \cdot 0 = \hat{M}_{4,3}$ so its size does not grow.

By applying **line 17** for $j = 4$, we have

$$\begin{aligned}
|M_{1,4}| = |\hat{M}_{1,4} - gM_{1,2}| &\leq |\hat{M}_{1,4}| + |g| \cdot |M_{1,2}| \\
&\leq \frac{8(361p + 642\sqrt{p} + 2568)}{p^5}\mathrm{nrd}(\alpha_1)^{19/2}\mathrm{nrd}(\beta_2)^{1/2}\frac{\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{5/2}}{\mathrm{nrd}(\beta_4)^{1/2}} \\
&\quad + \frac{8(361p + 642\sqrt{p} + 2568)}{p^5}\mathrm{nrd}(\alpha_1)^{19/2}\mathrm{nrd}(\beta_2)^{1/2}\frac{\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{5/2}}{\mathrm{nrd}(\beta_4)^{1/2}} \\
&\quad \times \frac{164^2(9p + 16\sqrt{p} + 64)^2}{p^{11}}\mathrm{nrd}(\alpha_1)^{20}\mathrm{nrd}(\beta_1)^{1/2}\mathrm{nrd}(\beta_2)^{3/2}\frac{\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{5}}{\mathrm{nrd}(\beta_4)} \\
&\leq \frac{8(361p + 642\sqrt{p} + 2568)}{p^5 \cdot p^9}\mathrm{nrd}(\alpha_1)^{19/2}\mathrm{nrd}(\beta_2)^{1/2}\frac{\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{15/2}}{\mathrm{nrd}(\beta_4)^{3/2}} \\
&\quad + \frac{8 \cdot 164^2(361p + 642\sqrt{p} + 2568)(10p)^2}{p^{16}}\mathrm{nrd}(\alpha_1)^{59/2}\mathrm{nrd}(\beta_1)^{1/2}\mathrm{nrd}(\beta_2)^{2}\frac{\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{15/2}}{\mathrm{nrd}(\beta_4)^{3/2}} \\
&\leq \frac{8 \cdot (1640^2 + 1)(361p + 642\sqrt{p} + 2568)}{p^{14}}\mathrm{nrd}(\alpha_1)^{59/2}\mathrm{nrd}(\beta_1)^{1/2}\mathrm{nrd}(\beta_2)^{2}\frac{\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{15/2}}{\mathrm{nrd}(\beta_4)^{3/2}}
\end{aligned}$$

since $p^{10} > p^9 + 16\sqrt{p} + 64$ in SQIsign.

$M_{2,4} = \hat{M}_{2,4} - gM_{2,2} = \hat{M}_{2,4} - (\hat{M}_{2,4} - r) = r$ so the size of $M_{2,4}$ does not grow.
$M_{3,4} = \hat{M}_{3,4} - gM_{3,2} = \hat{M}_{3,4} - g \cdot 0 = \hat{M}_{3,4}$ so its size does not grow.
$M_{4,4} = \hat{M}_{4,4} - gM_{4,2} = \hat{M}_{4,4} - g \cdot 0 = \hat{M}_{4,4}$ so its size does not grow.

By applying **line 17** from $j = 5$ to $j = 16$, we have

$$\begin{aligned}
|M_{1,j}| = |\hat{M}_{1,j} - gM_{1,2}| &\leq |\hat{M}_{1,j}| + |g| \cdot |M_{1,2}| \\
&\leq \frac{640(9p + 16\sqrt{p} + 64)}{p^{11/2}}\mathrm{nrd}(\alpha_1)^{19/2}\mathrm{nrd}\left(\alpha_{\lceil \frac{j}{4} \rceil}\right)^{1/2}\mathrm{nrd}(\beta_{((j-1) \bmod 4)+1})^{1/2}\mathrm{nrd}(\beta_2)^{1/2}\frac{\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{5/2}}{\mathrm{nrd}(\beta_4)^{1/2}} \\
&\quad + \frac{640(9p + 16\sqrt{p} + 64)}{p^{11/2}}\mathrm{nrd}(\alpha_1)^{19/2}\mathrm{nrd}\left(\alpha_{\lceil \frac{j}{4} \rceil}\right)^{1/2}\mathrm{nrd}(\beta_{((j-1) \bmod 4)+1})^{1/2}\mathrm{nrd}(\beta_2)^{1/2}\frac{\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{5/2}}{\mathrm{nrd}(\beta_4)^{1/2}} \\
&\quad \times \frac{164^2(9p + 16\sqrt{p} + 64)^2}{p^{11}}\mathrm{nrd}(\alpha_1)^{20}\mathrm{nrd}(\beta_1)^{1/2}\mathrm{nrd}(\beta_2)^{3/2}\frac{\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{5}}{\mathrm{nrd}(\beta_4)} \\
&\leq \frac{640(9p + 16\sqrt{p} + 64)}{p^{11/2} \cdot p^9}\mathrm{nrd}(\alpha_1)^{19/2}\mathrm{nrd}\left(\alpha_{\lceil \frac{j}{4} \rceil}\right)^{1/2}\mathrm{nrd}(\beta_{((j-1) \bmod 4)+1})^{1/2}\mathrm{nrd}(\beta_2)^{1/2}\frac{\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{15/2}}{\mathrm{nrd}(\beta_4)^{3/2}} \\
&\quad + \frac{640 \cdot 164^2(9p + 16\sqrt{p} + 64)(10p)^2}{p^{33/2}}\mathrm{nrd}(\alpha_1)^{59/2}\mathrm{nrd}\left(\alpha_{\lceil \frac{j}{4} \rceil}\right)^{1/2}\mathrm{nrd}(\beta_1)^{1/2}\mathrm{nrd}(\beta_2)^{2}\mathrm{nrd}(\beta_{((j-1) \bmod 4)+1})^{1/2}\frac{\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{15/2}}{\mathrm{nrd}(\beta_4)^{3/2}} \\
&\leq \frac{640 \cdot (1640^2 + 1)(9p + 16\sqrt{p} + 64)}{p^{29/2}}\mathrm{nrd}(\alpha_1)^{59/2}\mathrm{nrd}\left(\alpha_{\lceil \frac{j}{4} \rceil}\right)^{1/2}\mathrm{nrd}(\beta_1)^{1/2}\mathrm{nrd}(\beta_2)^{2}\mathrm{nrd}(\beta_{((j-1) \bmod 4)+1})^{1/2}\frac{\prod_{l=1}^{4}\mathrm{nrd}(\beta_l)^{15/2}}{\mathrm{nrd}(\beta_4)^{3/2}}
\end{aligned}$$

since $10p > 9p + 16\sqrt{p} + 64$ in SQIsign.

$M_{2,j} = \hat{M}_{2,j} - gM_{2,2} = \hat{M}_{2,j} - (\hat{M}_{2,j} - r) = r$ so the size of $M_{2,j}$ does not grow.
$M_{3,j} = \hat{M}_{3,j} - gM_{3,2} = \hat{M}_{3,j} - g \cdot 0 = \hat{M}_{3,j}$ so its size does not grow.
$M_{4,j} = \hat{M}_{4,j} - gM_{4,2} = \hat{M}_{4,j} - g \cdot 0 = \hat{M}_{4,j}$ so its size does not grow.

When we compute the loop of $i = 1$, each entry changes to the entry of HNF matrix.

By the result of the computation, we have a bound for the size of integers during the process of **HNF** algorithm as

$$\frac{4000 \cdot (1640+1)^2}{p^{25/2}} \max_{1\le s\le 4} \mathrm{nrd}(\alpha_s)^{30} \max_{1\le t\le 4} \mathrm{nrd}(\beta_t)^7 \prod_{l=1}^{4} \mathrm{nrd}(\beta_l)^6$$

since $10p > 9p + 16\sqrt{p} + 64$ and $10p^{3/2} > 9p^{3/2} + 16p + 64$ and $\frac{6400}{4000} < p$ in SQIsign.
After the HNF algorithm, we have

$$|M_{i,j}| \le 2^{10} \left(1+\frac{1}{p}\right)^4 \max_{1\le s\le 4} \left(\mathrm{nrd}(\alpha_s)^2\right) \max_{1\le t\le 4} \left(\mathrm{nrd}(\beta_t)^2\right)$$

by Lemma 33.

Hence, when $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}, \{\beta_1, \beta_2, \beta_3, \beta_4\}$ are shortest bases, while applying **HNF** algorithm, we have

$$\begin{aligned} |M_{i,j}| &\le \frac{4000 \cdot (1640^2+1)}{p^{25/2}} \left(\frac{2^{10}}{\pi^4} \det(\mathcal{O}_0)^2 \mathrm{nrd}(I_1)^4\right)^{30} \left(\frac{2^{10}}{\pi^4} \det(\mathcal{O}_0)^2 \mathrm{nrd}(I_2)^4\right)^{19/2} \\ &= \frac{4000 \cdot (1640^2+1)}{p^{25/2}} \frac{2^{395}}{\pi^{158}} \left(\det(\mathcal{O}_0)^{79} \mathrm{nrd}(I_1)^{120} \mathrm{nrd}(I_2)^{38}\right) \end{aligned}$$

by Lemma 34. □

Lemma 33. *Let $I_1 = \langle \alpha_1, \alpha_2, \alpha_3, \alpha_4 \rangle, I_2 = \langle \beta_1, \beta_2, \beta_3, \beta_4 \rangle$ be two left $\mathcal{O}_0$-ideals. Then, for $I_1 I_2 = \langle \gamma_1, \gamma_2, \gamma_3, \gamma_4 \rangle$, we have*

$$|a_{\gamma_l}|, |b_{\gamma_l}|, |c_{\gamma_l}|, |d_{\gamma_l}| \le 2^{10} \left(1+\frac{1}{p}\right)^4 \max_{1\le s\le 4} \left(\mathrm{nrd}(\alpha_s)^2\right) \max_{1\le t\le 4} \left(\mathrm{nrd}(\beta_t)^2\right)$$

Proof. By Lemma 1, we have

$$|a_{\alpha_s\beta_t}|, |b_{\alpha_s\beta_t}| \le 4\sqrt{\mathrm{nrd}(\alpha_s)\mathrm{nrd}(\beta_t)}$$

$$|c_{\alpha_s\beta_t}|, |d_{\alpha_s\beta_t}| \le \frac{4}{\sqrt{p}} \sqrt{\mathrm{nrd}(\alpha_s)\mathrm{nrd}(\beta_t)}$$

So for any column $\mathfrak{c}_{s,t}$ of $\tilde{M}$ corresponding to the vector consisting of coefficients of $\alpha_s\beta_t$, we have

$$\|\mathfrak{c}_{s,t}\|_\infty = \max\{|a_{\alpha_s\beta_t}|, |b_{\alpha_s\beta_t}|, |c_{\alpha_s\beta_t}|, |d_{\alpha_s\beta_t}|\} \le 4\sqrt{\mathrm{nrd}(\alpha_s)\mathrm{nrd}(\beta_t)}$$

$$\|\mathfrak{c}_{s,t}\|_2 = \sqrt{|a_{\alpha_s\beta_t}|^2 + |b_{\alpha_s\beta_t}|^2 + |c_{\alpha_s\beta_t}|^2 + |d_{\alpha_s\beta_t}|^2} \le 4\sqrt{\left(2+\frac{2}{p}\right)\mathrm{nrd}(\alpha_s)\mathrm{nrd}(\beta_t)}$$

Now let $\mathfrak{M}$ be any arbitrary 4×4 submatrix of $\tilde{M}$. Then, by Hadamard's inequality, we have

$$|\det(\mathfrak{M})| \leq \prod_{l=1}^{4} \|\mathfrak{M}_l\|_2$$

where $\mathfrak{M}_l$ is the l-th column of $\mathfrak{M}$.
Since we have $\|\mathfrak{M}_l\|_2 \leq 4\sqrt{\left(2+\frac{2}{p}\right)\mathrm{nrd}(\alpha_s)\mathrm{nrd}(\beta_t)}$,
we get

$$\begin{aligned}|\det(\mathfrak{M})| &\leq \left(4\sqrt{\left(2+\frac{2}{p}\right)\mathrm{nrd}(\alpha_s)\mathrm{nrd}(\beta_t)}\right)^4 \\ &= 2^{10}\left(1+\frac{1}{p}\right)^4 \max_{1\leq s\leq 4}\left(\mathrm{nrd}(\alpha_s)^2\right)\max_{1\leq t\leq 4}\left(\mathrm{nrd}(\beta_t)^2\right).\end{aligned}$$

Let $\Delta = \gcd\{\det(\mathfrak{M}) \mid \mathfrak{M} \text{ is a } 4\times 4 \text{ submatrix of } \tilde{M}\}$.

Let $\mathcal{M}$ be a 4×4 integer matrix whose each column consists of coefficients of γ_l. Then, we have $\det(\mathcal{M}) = \Delta$ by [38]. Since $\mathcal{M}$ is a triangular integer matrix,

$$|\mathcal{M}_{s,t}| \leq \max_{1\leq r\leq 4}|\mathcal{M}_{r,r}| \leq \Delta$$

□

Lemma 34. *Let $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ be a shortest basis of a left $\mathcal{O}_0$-ideal I. Then, for any $1 \leq l \leq 4$,*

$$\mathrm{nrd}(\alpha_l) \leq \frac{32}{\pi^2}\det(\mathcal{O}_0)\mathrm{nrd}(I)^2$$

and

$$\prod_{r=1}^{4}\mathrm{nrd}(\alpha_r) \leq \frac{32}{\pi^2}\det(\mathcal{O}_0)\mathrm{nrd}(I)^2$$

Proof. Let $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}$ be a shortest basis of a left $\mathcal{O}_0$-ideal I and λ_i be the i-th successive minima, where $\|\alpha_i\|_{\mathbb{Z}} = \lambda_i$ for $i = 1, 2, 3, 4$, W.L.O.G. (We can apply the result by rearranging the basis.)
By Minkowski's First Theorem, we have $\lambda_1 \leq \frac{2^{5/4}}{\sqrt{\pi}}\det(I)^{1/4}$. So we have $\mathrm{nrd}(\alpha_1) = \|\alpha_1\|_{\mathbb{Z}}^2 \leq \left(\frac{2^{5/4}}{\sqrt{\pi}}\det(I)^{1/4}\right)^2$.
Since $\det(I)^{1/2} = \det(\mathcal{O}_0)^{1/2}\mathrm{nrd}(I)$ by the proof of proposition 2.11 in [39], we have

$$\mathrm{nrd}(\alpha_1) \leq \frac{2^{5/2}}{\pi}\det(\mathcal{O}_0)^{1/2}\mathrm{nrd}(I)$$

By Minkowski's Second Theorem, we have

$$\prod_{i=1}^{4} \lambda_i \leq \frac{32}{\pi^2} \det(I) = \frac{32}{\pi^2} \det(\mathcal{O}_0)\mathrm{nrd}(I)^2.$$

Since I is a left $\mathcal{O}_0$-ideal, it is clear that $\lambda_i \geq 1$ for each i. So we get $\lambda_4 \leq \frac{32}{\pi^2} \det(\mathcal{O}_0)\mathrm{nrd}(I)^2$. Hence,

$$\mathrm{nrd}(\alpha_l) \leq \mathrm{nrd}(\alpha_4) = \lambda_4^2 \leq \frac{2^{10}}{\pi^4} \det(\mathcal{O}_0)^2\mathrm{nrd}(I)^4.$$

□

G Statements cited from references

We summarize some statements cited from other references, which we use in the proofs of some lemmas.

1. Exercise 1.7.7.(4) of [29]
   For a unitary matrix U, a map $T \mapsto UTU^*$ is an isometry.

2. A statement in the proof of 1.1.(vi) in [30]
   If A is a positive definite $n \times n$ matrix over $\mathbb{R}$, then for each $x, y \in \mathbb{R}^n$,
   $$\langle x, Ay\rangle_2 = \langle \sqrt{A}x, \sqrt{A}y\rangle_2$$

3. Proposition 13.3.4 of [18]
   Let D be a division algebra over a field F and R be a ring of integers of F, and $\mathcal{O} = \{\alpha \in D \mid w(\alpha) \geq 0\}$ where w is the extended valuation of D. The ring $\mathcal{O}$ is the unique maximal R-order in D, consisting all elements of D that are integral over R.

4. Lemma 17.4.6 of [18]
   The orders $\mathcal{O}, \mathcal{O}'$ are connected if and only if $\mathcal{O}, \mathcal{O}'$ are isomorphic.

5. Proposition 16.6.15.(a) of [18]
   Let I be an ideal in $B_{p,\infty}$. Then, $I\bar{I} = \mathrm{nrd}(I)O_L(I)$ and $\bar{I}I = \mathrm{nrd}(I)O_R(I)$.

6. A statement in the proof of Lemma 48 in [26]
   When $(\alpha_1, \ldots, \alpha_4)$ is a Minkowski-reduced basis of a left $\mathcal{O}$-ideal I for a maximal order $\mathcal{O}$ in $B_{p,\infty}$, we have
   $$\prod_{i=1}^{4} \|\alpha_i\| \leq \frac{64}{\pi^2} p^2$$
   where p is the parameter in SQIsign.

7. Theorem 16.1.3(iv),(iv') of [18] For a given ideal I in $B_{p,\infty}$,

$$\mathrm{nrd}(I)^2 = [O_L(I) : I] = [O_R(I) : I]$$

8. A.1.9 of [33]
Let $\{\mathcal{C}_i \mid i \in I\}$ be a set of objects of $\mathcal{C}$. Then, there is a object $\prod_{i\in I} \mathcal{C}_i$ of $\mathcal{C}$, with morphisms $\pi_j : \prod_{i\in I} C_i \to \mathcal{C}_j$ such that for all family of morphisms $\{\alpha_i\} : A \to \mathcal{C}_i$, there is a unique morphism $\alpha : A \to \prod_{i\in I}$.

9. Proposition 6.9 of [34]
The function l which maps a module to its length, is an additive function.

10. 2.4.13.(2) of [35]
Let L be a $\mathbb{Z}$-submodule of a free module L' and of the same rank. There exist positive integers $d_1, \ldots, d_n$ such that $L'/L \cong \bigoplus_{i=1}^{n} \mathbb{Z}/d_i\mathbb{Z}$ and $[L' : L] = \prod_{i=1}^{n} d_i$.

11. Lemma 15.2.15 of [18]
Let I, J be projective lattices in $B_{p,\infty}$. Then,

$$\mathrm{disc}(I) = [J : I]^2_{\mathbb{Z}} \mathrm{disc}(J).$$

12. 1.11 of [31]
Let B be a matrix representing the basis of a lattice. Then,

$$\det(\mathcal{L}(B)) = \sqrt{\det(B^T B)}.$$