

The Lattice Isomorphism Problem with Hints

Mélissa Rossi

CryptoExperts, Paris, France
melissa.rossi@cryptoexperts.com

Abstract. The Lattice Isomorphism Problem (LIP) is a relatively new problem that has gained increasing attention in the field of lattice-based cryptography. It is the underlying hard problem of the HAWK signature scheme, which has been submitted to the second NIST call for post-quantum signatures. In this paper, we propose to study the security of LIP in the presence of partial information on the secret solution, which can be obtained through side-channel attacks or by design. We propose a new cryptanalysis approach for LIP that embeds it into a shortest vector problem (SVP) based on a different lattice than the one considered in the specifications of HAWK, allowing to integrate hints. We also introduce conversion techniques that transform hints expressed on arbitrary linear combinations of the secret key coefficients into hints on the target lattice, enabling the integration of a broader class of linear leakages on sensitive values. We implement our attack and predictions using the LeakyLWEEstimator tool (Crypto 2020) and obtain concrete security estimates for HAWK in the key-recovery setting. We also observe that HAWK signatures carry intrinsic information about the secret key and show that this can be exploited as a source of hints, yielding a refined key-recovery complexity, thereby eroding the original security estimates. For example, we estimate a security loss of around 11 bits for HAWK-1024. Finally, we apply our framework to side-channel analysis results on HAWK, solving an open problem raised in a recent analysis (TCHES 2024) and discuss further attack paths enabled by fault-injection techniques. We believe that this new cryptanalysis approach for LIP and the integration of hints provide new insights towards a better understanding of its security.

Keywords: Lattice Isomorphism Problem · HAWK signature scheme · Hints

1 Introduction

Lattice-based cryptography is a central area of post-quantum cryptography. The security of schemes in this family relies on the hardness of computational problems over lattices such Learning With Errors (LWE) and Short Integer Solution (SIS). Three post-quantum NIST standards are lattice-based: the encryption scheme ML-KEM [SAB⁺22], and the signature schemes ML-DSA [LDK⁺22] and FN-DSA [PFH⁺22].

These schemes are dimensioned to resist the best known attacks in a black-box security model where an adversary only has access to input-output behaviour. Such attacks rely on lattice reduction algorithms, whose complexity has been actively studied over the past decade, see e.g. [ADPS16, AGVW17]. Although their precise performance is not fully understood, predictions for the state-of-the-art lattice reduction algorithm BKZ [SE94, CN11] have stabilized in recent years. These predictions are implemented in tools such as the LWE estimator [ACD⁺18], which provides concrete security estimates.

Another line of research goes beyond black-box attacks by studying security in the presence of side information. Such information may arise from side-channel attacks,

which exploit physical leakages of an implementation, or from structural properties of the scheme (e.g. partial knowledge of the Hamming weight of the secret key). These leakages, referred to as *hints*, can be incorporated into attacks to reduce their complexity. The study of side-channel attacks on lattice-based schemes has intensified in recent years, with numerous works demonstrating key recovery under various leakage models [KH18, MUTS22, GMRR22, ZLYW23]. This has led to increased interest in the integration of hints into security analyses [DDGR20, DGHK23, MN23], as well as the development of dedicated tools such as the LeakyLWEEstimator [DDGR20].

HAWK [BBD⁺23a] is a signature scheme submitted to the second NIST call for post-quantum signatures. It achieves very competitive performance, surpassing FN-DSA in some settings, by relying on a different structured lattice problem: the Lattice Isomorphism Problem (LIP) [HR14, DPPvW22, DvW22]. This problem is relatively recent compared to classical lattice problems. On the theoretical side, LIP and its variants have attracted increasing attention [MPPW24, LJPW24, APv25, CME⁺25]. However, from a concrete security estimation perspective, integrating LIP into lattice reduction frameworks is less straightforward than for SIS and LWE problems. At the time of the initial version of this work¹, the only concrete estimation was proposed by the designers of HAWK using an analysis of [DDGR20], but the prediction had to be adapted to handle the specificities of the LIP problem where the underlying structure is an isomorphism class.

HAWK has also been the target of side-channel analyses. While no countermeasures are currently implemented, the reference implementation has been studied and several leakage sources have been identified, in particular in the Gaussian sampler [GR24]. These attacks are highly effective: about 98% of the secret key can be recovered, and the remaining uncertainty can be resolved with limited brute force (the authors estimate about 2^{55} elementary operations). The authors suggested incorporating these hints into the LeakyLWEEstimator, but left this as an open problem because of the difficulty of integrating them into the current cryptanalytic framework. Indeed, the state of the art for estimating the best attack on HAWK does not apply lattice reduction to a lattice basis, but rather to an isomorphism class of lattices [BBD⁺23b], and is therefore inherently incompatible with hints. Resolving this issue is highly relevant, because in more realistic protected implementations the recovered fraction of the secret key will quickly become too small for brute force to remain feasible. Quantifying the impact on the bit security in a granular way makes it possible to evaluate realistic implementations and to account for inaccuracies in leakage extraction. More broadly, a better understanding of the impact of the LIP structure, which differs from SIS and LWE, on the lattice reduction complexity also contributes to a more refined security analysis of HAWK, even in the absence of side-channel information.

1.1 Contributions

The initial goal of this work was to address the open problem left in [GR24], namely how to integrate the available hints into a security estimator. Achieving this actually required first resolving the challenge of directly estimating the hardness of LIP via lattice reduction, and then adapting the framework to the specific structure of the available hints. This approach also yielded new insights into the security of HAWK even in the absence of side-channel information. Our contributions are as follows:

- **A new SVP-based cryptanalysis of LIP.** First, we reuse the SVP reduction of [CME⁺25] and combine it with a new lattice distortion technique that yields new key-recovery estimates, beating the previous estimates for HAWK-1024. This

¹Later on, more cryptanalysis have been proposed [SW26, MPM26], see the future work section.

approach is different from the one considered in the specification [BBD⁺23b], and allows hints to be incorporated in a standard way, directly compatible with the LeakyLWEEstimator framework.

- **A hint conversion framework.** We develop techniques to convert hints expressed as arbitrary linear combinations of all secret key coefficients into equivalent hints on the target lattice, thereby enabling the integration of a broader class of linear leakages on all sensitive values.
- **Integration into the LeakyLWEEstimator and new key recovery security estimates.** We implement our attack adapting the LeakyLWEEstimator and derive first concrete key-recovery security estimates for HAWK. Our results are consistent with, and in some cases improve upon, those reported in the specification.
- **Signatures as a source of hints.** We show that HAWK signatures inherently leak information about the secret key and demonstrate how this can be exploited as a source of hints, leading to a measurable reduction in security. This shows that the signature parameter choices, particularly the average and maximum sample norm, have a significant impact on the key-recovery security level, and that this intrinsic leakage erodes the security margin.
- **Applications to side-channel attacks.** Returning to the original motivation of this work, we apply our framework to the side-channel results on HAWK [GR24], resolving the open problem identified in that work and outlining potential attack vectors enabled by fault injection.

Organization of the paper. In Section 2, we review the necessary background on lattices, the LIP problem, and the HAWK scheme. In Section 3, we present our SVP embedding of LIP, followed by the integration of hints in Section 4. In Section 5, we describe the implementation within the LeakyLWEEstimator and provide concrete security estimates. Applications, including the use of signatures as hints and side-channel analysis, are discussed in Section 6.

2 Preliminaries

2.1 Algebra

Notations. Vectors will be denoted with bold lowercase letters. We denote by $\|\mathbf{v}\|$ the Euclidean norm L_2 of a vector $\mathbf{v}$ and by $\langle \mathbf{u}, \mathbf{v} \rangle$ its associated inner product with a second vector $\mathbf{u}$. By default, all vectors are column vectors. The transpose of a matrix or a vector $\mathbf{M}$ is denoted by $\mathbf{M}^T$.

Number fields. Let n be a power of two, $K_n = \mathbb{Q}[X]/(X^n + 1)$ a cyclotomic number field and its ring of integers $R_n = \mathbb{Z}[X]/(X^n + 1)$. For a polynomial $f \in K_n$, let $f = f_0 + f_1X + \dots + f_{n-1}X^{n-1}$ and define a bijection that maps it to the column vector of its coefficients $\text{vec} : K_n \rightarrow \mathbb{Q}^n, f \mapsto (f_0, \dots, f_{n-1})$. We denote by $\vec{f} := [f_0, \dots, f_{n-1}]^T$ the column vector of the coefficients of f .

Hermitian adjoint and Gram matrix. We denote the Hermitian adjoint of an element $f \in K_n$ by $f^* = f_0 - \sum_{i=1}^{n-1} f_i X^i$. We extend the notations to elements $\mathbf{B} \in K_n^{r \times r}$: $\mathbf{B}^*$ denotes the entry-wise adjoint of $\mathbf{B}^T$. For any element $\mathbf{B} \in K_n^{r \times r}$, we call the Gram matrix of $\mathbf{B}$ the matrix $\mathbf{Q} = \mathbf{B}^* \cdot \mathbf{B}$. We denote by $\|\cdot\|_{\mathbf{Q}}$ the norm associated to the Gram matrix $\mathbf{Q}$.

Negacyclic matrix. We call the negacyclic matrix of $f \in K_n$ the matrix defined as follows:

$$\boxed{f}^\rightarrow := \begin{pmatrix} f_0 & \cdots & f_{n-1} \\ \vdots & \ddots & \vdots \\ -f_{n-1} & \cdots & f_0 \end{pmatrix}.$$

Note that the negacyclic matrix satisfies the following properties that will be used throughout the paper. Let $f, g \in K_n$, we have:

1. $\boxed{f}^\rightarrow \cdot \vec{g} = \overrightarrow{f \cdot g} \bmod X^n + 1$
2. $\boxed{f}^\rightarrow \cdot \boxed{g}^\rightarrow = \boxed{(f \cdot g \bmod X^n + 1)}^\rightarrow$.
3. for f invertible,
 $\boxed{f}^{\rightarrow^{-1}} = \boxed{f^{-1} \bmod X^n + 1}^\rightarrow$.
4. $\boxed{f}^{\rightarrow T} = \boxed{f^*}^\rightarrow$.
5. $f \cdot f^* = \sum_{i=0}^{n-1} f_i^2 = \|\vec{f}\|^2$.
6. $\vec{f}^{\rightarrow T} \cdot \vec{g}^\rightarrow = \vec{f}^T \cdot \vec{g}$.

Specific matrix. Let n be an integer. Let $\mathcal{GL}_n(\mathbb{Z})$ be the set of unimodular matrix of $\mathbb{Z}^{n \times n}$. A matrix $\mathbf{U} \in \mathbb{Z}^{n \times n}$ is a unimodular matrix if $\det(\mathbf{U}) = \pm 1$. Besides, let $\mathcal{O}_n(\mathbb{R})$ be the orthogonal group of $\mathbb{R}^n$. Concretely, $\mathcal{O}_n(\mathbb{R}) = \{\mathbf{O} \in \mathbb{R}^{n \times n} \mid \mathbf{O}^T \mathbf{O} = \mathbf{I}_n\}$ where $\mathbf{I}_n$ is the identity matrix of size n .

2.2 Lattices

Let $m, n \in \mathbb{N}$ such that $m \geq n$. A lattice Λ is the set of all integer linear combinations of n linearly independent vectors $\{\mathbf{b}_j\} \subset \mathbb{R}^m$. The integer n is called the rank of the lattice and m is called the dimension. Formally, we have

$$\Lambda = \left\{ \sum_{i=1}^n z_i \mathbf{b}_i \mid z_i \in \mathbb{Z} \right\}.$$

Usually the basis vectors are integers, *i.e.* $\mathbf{b}_i \in \mathbb{Z}^m$. However in this paper, we stick to the more general definition where $\mathbf{b}_i \in \mathbb{R}^m$ because it fits better with the LIP problem. Note that the z_i remain integers.

A matrix whose columns are these basis vectors is also called a basis. Note that many bases generate the same lattice. More precisely, if $\mathbf{B}$ is a basis of a lattice Λ , then for any unimodular matrix $\mathbf{U} \in \mathbb{Z}^{n \times n}$, the matrix $\mathbf{B} \cdot \mathbf{U}$ is also a basis of Λ . In this paper, we will informally say that some bases are "good bases" if they contain short vectors, and "bad bases" otherwise. The volume of a lattice is defined as:

$$\text{Vol}(\Lambda) = \sqrt{\det(\mathbf{B}^T \mathbf{B})}$$

and is independent from the choice of the basis. The volume can be seen as a measure of the density of the lattice. The smaller the volume, the denser the lattice.

One of the main problems when working with lattices is the Shortest Vector Problem (SVP), which consists in finding a non-zero vector of minimal length in a given lattice Λ . Formally, SVP is defined as follows:

Hard problem 1 (Shortest Vector Problem (SVP) [LJL82, Ajt96]). Given a basis $\mathbf{B}$ of a lattice Λ , find a non-zero vector $\mathbf{s} \in \Lambda$ such that $\|\mathbf{s}\| = \lambda_1(\Lambda)$ where $\lambda_1(\Lambda) = \min_{\mathbf{s} \in \Lambda \setminus \{0\}} \|\mathbf{s}\|$.

This hard problem is at the core of all lattice-based schemes. When given a basis without short vectors, the best algorithm for solving this problem has an exponential

complexity and is called BKZ [SE94]. BKZ requires a block size parameter β . The final complexity of BKZ is exponential in β , the precise conversion from block size to bit security is not known but it is estimated in cost models. We refer to [CN11] for more details on the complexity of BKZ. Usually, the $0.292 \times \beta$ cost model of [ACD⁺18] is used.

The main metric for estimating the difficulty of an SVP instance is the required BKZ block size β . The usual prediction of the required β to solve an SVP instance with $\|\mathbf{s}\|$ being the norm of a shortest vector and with BKZ- β is given by the inequality [ADPS16, AGVW17]:

$$\sqrt{\beta / \dim(\Lambda)} \cdot \|\mathbf{s}\| \leq \delta_\beta^{2\beta - \dim(\Lambda) - 1} \cdot \text{Vol}(\Lambda)^{1/\dim(\Lambda)} \quad (1)$$

where δ_β is called the root Hermite factor of BKZ- β . For $\beta < 50$, the root Hermite factor is tabulated [ACD⁺18]. For $\beta \geq 50$, the root Hermite factor is predictable using the Gaussian Heuristic [CN11]:

$$\delta_\beta = \left((\pi\beta)^{1/\beta} \cdot \frac{\beta}{2\pi e} \right)^{1/(2\beta-2)}.$$

2.3 Lattice Isomorphism Problem (LIP)

Two lattices Λ and Λ' are said isomorphic if there exists an orthonormal transformation $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$ such that $\mathbf{O} \cdot \Lambda = \Lambda'$. The problem of finding such an isomorphism is called the Lattice Isomorphism Problem (LIP) [HR14].

Hard problem 2 (The original Lattice Isomorphism Problem [HR14]). Given two isomorphic lattices Λ and Λ' , the Lattice Isomorphism Problem (LIP) consists in finding $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$ such that $\mathbf{O} \cdot \Lambda = \Lambda'$.

When using lattices, the most convenient is to handle their bases. However, as many bases can generate the same lattice, an instantiation of the LIP problem in terms of basis needs the introduction of a third matrix to go from the basis of Λ , say $\mathbf{B}$, to the basis of Λ' , say $\mathbf{B}'$. More precisely, solving the LIP problem remains to show that there exists an orthonormal matrix $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$ and a unimodular matrix $\mathbf{U} \in \mathcal{GL}_n(\mathbb{Z})$ such that $\mathbf{O} \cdot \mathbf{B} \cdot \mathbf{U} = \mathbf{B}'$. Hence, to avoid handling the orthonormal matrix explicitly, one can equivalently define the LIP problem with the notion of Gram matrix.

Hard problem 3 (Lattice Isomorphism Problem with quadratic forms [DvW22]). Given two isomorphic lattices Λ and Λ' with Gram matrix $\mathbf{Q}$ and $\mathbf{Q}'$, the Lattice Isomorphism Problem (LIP) consists in finding a unimodular matrix $\mathbf{U} \in \mathcal{GL}_n(\mathbb{Z})$ such that $\mathbf{Q}' = \mathbf{U}^T \mathbf{Q} \mathbf{U}$.

To get to the problem that is actually used in HAWK, one needs to use a generalization of LIP to module lattices.

Definition 1 (Module lattice [LPR10]). A module lattice over K_n of rank r is a lattice Λ such that its basis $\mathbf{B}$ is an element of $K_n^{r \times r}$. More precisely, the basis $\mathbf{B}$ is an integer matrix of size $rn \times rn$ composed of r^2 negacyclic blocks of size $n \times n$.

Hard problem 4 (Worst case smLIP [DPPvW22, DvW22]). Given any Gram matrix $\mathbf{Q}'$ $\mathbf{Q}$ of module lattices of rank r over K_n , find $\mathbf{U} \in \mathcal{GL}_r(R_n)$ such that $\mathbf{Q}' = \mathbf{U}^* \mathbf{Q} \mathbf{U}$.

The authors of HAWK [DPPvW22] chose to consider a particular case of this problem with $r = 2$ and $\mathbf{Q} = \mathbf{I}$. Interestingly, one can even see this particular case as a factorization problem.

Hard problem 5 (Particular case of Module-LIP as a factorization problem [DPPvW22, DvW22]). Given any Gram matrix $\mathbf{Q}$ of a module lattice of rank 2 over K_n , find $\mathbf{B} \in \mathcal{GL}_2(R_n)$ such that $\mathbf{Q} = \mathbf{B}^* \mathbf{B}$.

Note that it is always possible to find a factorization by computing a Cholesky decomposition of $\mathbf{Q}$ in polynomial time. However, the challenge is to find a solution $\mathbf{B}$ with *integer* coefficients. Note as well that none of these definitions use the fact that there are small values, i.e. short vectors to recover. Indeed, we will see in Section 3.1 that the link to SVP (Hard problem 1) is actually not as direct as for well-known lattice-based problems such as SIS or LWE.

2.4 Hawk signature scheme

We describe here the high-level idea behind HAWK. For further details, we refer to the official specification [BBD⁺23a]. Let $n \in \{256, 512, 1024\}$ be a fixed parameter.

Hawk keypair. The HAWK private key consists of four small polynomials $f, g, F, G \in R_n$ which satisfy the equation $fG - gF = 1 \bmod X^n + 1$. Note that $\mathbf{B} := \begin{pmatrix} f & F \\ g & G \end{pmatrix}$ is then a unimodular matrix over $R_n^{2 \times 2}$ since its determinant is equal to 1 mod $X^n + 1$. It is the solution of Hard problem 5 with the following Gram matrix

$$\mathbf{Q} := \mathbf{B}^* \cdot \mathbf{B} = \begin{pmatrix} q_{00} & q_{01} \\ q_{01}^* & q_{11} \end{pmatrix} = \begin{pmatrix} f^2 + g^2 & f^*F + g^*G \\ F^*f + G^*g & F^2 + G^2 \end{pmatrix}. \quad (2)$$

In addition, the polynomials f and g are sampled according to a centered binomial distribution with $\eta = 2, 4$ or 8 for $n = 256, 512$ or 1024 , respectively. The algorithm NTRUSolve [DLP14, PFH⁺22] is then used to find small Bézout coefficients F and G to verify the previous equation. Lastly, the keypair (sk, pk) is defined as $sk := \mathbf{B}$ and $pk := \mathbf{Q}$. Note that both matrix are $R_n^{2 \times 2}$, meaning that they can be represented with $4n$ coefficients in $\mathbb{Z}$.

Signature overview. At a high level, the HAWK signature algorithm proceeds as follows:

1. Hash the message to a vector $\mathbf{h} \in R_n^2$.
2. Compute a binary target $\mathbf{t} = \mathbf{B}\mathbf{h} \bmod 2$ using the private key $\mathbf{B}$.
3. Sample a vector $\mathbf{x}$ from a distribution that approximates a Gaussian on $2\mathbb{Z}^{2n} + \mathbf{t}$ with a standard deviation of $2\sigma_{\text{sign}}$ where σ_{sign} is a fixed parameter. Hence, $\|\mathbf{x}\|$ is small.
4. Compute $\mathbf{w} = \mathbf{B}^{-1}\mathbf{x}$ and then the signature $\text{sig} := \frac{1}{2}(\mathbf{h} - \mathbf{w})$.
5. Verify that the norm $\|\mathbf{w}\|_{\mathbf{Q}} = \|\mathbf{x}\|$ is not larger than $2\sigma_{\text{ver}}\sqrt{2n}$ where $\sigma_{\text{ver}} > \sigma_{\text{sign}}$ is a fixed parameter.

To verify a signature, $\mathbf{w}$ is recomputed from public data (sig and $\mathbf{h}$) and $\|\mathbf{w}\|_{\mathbf{Q}} = \|\mathbf{x}\|$ is checked to be short enough.

Discrete sampling. To sample a short vector $\mathbf{x}$ in step 3, the signature generation uses a discrete Gaussian sampler. The center is the target $\mathbf{t}$ and the standard deviation is fixed for each variant of HAWK. Each coefficient $\mathbf{x}[i]$ is sampled separately, with center being the corresponding $\mathbf{t}[i]$. The distribution of $\mathbf{x}$ without the knowledge of $\mathbf{t}$ is then a mixture of discrete Gaussians centered on the $2\mathbb{Z}^{2n} + \mathbf{t}$ cosets, that can be approximated by a multivariate discrete Gaussian with standard deviation $2\sigma_{\text{sign}}$.

We next introduce a property that shows that the security of HAWK relies on finding only one among the four secret polynomials f, g, F, G .

Property 1 (Extension of [MPPW24]). Let n be a parameter of HAWK. Given a HAWK public key $\mathbf{Q}$ and any one of the four secret polynomials f, g, F, G , there exists a quantum algorithm that recovers an equivalent secret key. Its running time is polynomial in the input size, apart from a factor exponential in r , where r is the number of distinct prime ideal factors of the principal ideal generated by the corresponding element among $q_{00} - ff^*$, $q_{00} - gg^*$, $q_{11} - FF^*$, and $q_{11} - GG^*$, depending on the given secret polynomial. In particular, the algorithm runs in quantum polynomial time when r is bounded by a constant.

Proof. Without loss of generality, suppose that f is given. The complementary secret polynomial g satisfies

$$g \cdot g^* = q_{00} - f \cdot f^*.$$

It is non-trivial to recover g from this equation, as it requires solving a norm equation in the cyclotomic field K_n with respect to the adjoint operation $*$. After factoring the absolute integer norm of $q_{00} - f \cdot f^*$ using Shor's algorithm [Sho94], the algorithm of [MPPW24, Theorem 2.16] returns the list of all solutions g' to this norm equation. This list contains the original g . Each candidate g' can then be tested in polynomial time for compatibility with the full public key $\mathbf{Q}$. We retain any candidate that can be completed into a basis $\mathbf{B}'$ satisfying $\mathbf{B}'^* \mathbf{B}' = \mathbf{Q}$. Such a basis is an equivalent secret key, and at least one candidate passes this test because the original g belongs to the list. By [MPPW24, Theorem 2.16], the list size and the running time are exponential in the number r of distinct prime ideal factors of the principal ideal generated by $g \cdot g^*$, and are otherwise polynomial in the input size. The cases where g, F , or G is given are analogous. $\square$

3 A new approach for embedding LIP to SVP

In HAWK NIST reference documentation [BBD⁺23b], the technique to assess the complexity of recovering any of the f, g, F , or G requires lattice reduction algorithms directly on the Gram matrix $\mathbf{Q} \in \mathbb{Z}^{2n \times 2n}$. Any unimodular matrix $\mathbf{U} \in \text{GL}_{2n}(\mathbb{Z})$ such that $\mathbf{Q} = \mathbf{U}^T \mathbf{U}$ provides a secret key that is equivalent to the original one. Indeed, it allows to sign any message by using an unstructured variant of the signing algorithm. Hence, an equivalent key recovery for HAWK boils down to the following problem, that is an extension of **Hard problem 5** for unstructured matrix:

Hard problem 6 (Equivalent key recovery problem for HAWK). Given any Gram matrix $\mathbf{Q}$ of a module lattice of rank 2 over K_n as a matrix in $\mathbb{Z}^{2n \times 2n}$, find $\mathbf{B} \in \text{GL}_{2n}(\mathbb{Z})$ such that $\mathbf{Q} = \mathbf{B}^T \mathbf{B}$.

There exists variants of lattice reduction tools that directly apply in the Gram matrix of the lattice. Applying lattice reduction algorithms such as BKZ on the lattice defined by the basis $\mathbf{Q}$ allows to recover a unimodular matrix $\mathbf{U}$ such that $\mathbf{U}^T \mathbf{Q} \mathbf{U} = \mathbf{I}$. If we set $\mathbf{V} = \mathbf{U}^{-1}$, $\mathbf{V}$ is still unimodular and then we have $\mathbf{Q} = \mathbf{V}^T \mathbf{V}$. Hence, $\mathbf{V}$ is a valid

equivalent private key for HAWK solving **Hard problem 6**.

This approach reduces the Gram matrix to the identity, so the reduced basis is fixed and public; equivalently, it works only up to isomorphism. The shortest vector is therefore the known vector $(1, 0, \dots, 0)$ and all of its negacyclic rotations and the secret key is recovered only through the unimodular transformation and not through the shortest vector. Consequently, this method is not naturally compatible with hints, since the hints apply to the secret unimodular transformation rather than to a shortest vector in the reduced lattice. In the following, we present an alternative approach that offers greater flexibility for incorporating hints, as it will be detailed in **Section 4**.

3.1 Embedding LIP into a first SVP problem [CME⁺25]

We introduce two lattices with a good basis containing secret short vectors and the bad basis containing the public key. This lemma is coming from a remark made in the introduction of [CME⁺25].

Lemma 1 (Generalization of [CME⁺25]). *Let $\mathbf{Q} = \mathbf{B}^* \cdot \mathbf{B} = \begin{pmatrix} q_{00} & q_{01} \\ q_{01}^* & q_{11} \end{pmatrix} \in R_n^{2 \times 2}$ be a HAWK public key related to the private key $\mathbf{B} = \begin{pmatrix} f & F \\ g & G \end{pmatrix} \in R_n^{2 \times 2}$. The bases in the first line of **Table 1** generate the same lattice as the bases in the second line.*

Lemma 1 allows to separately target (f, g) and (F, G) and hence falling back into the setting where lattice reduction techniques target the actual secret. The proof is based on the fact that

$$\begin{pmatrix} I_n & 0_n \\ \overrightarrow{q_{01}^*} & \overrightarrow{q_{00}} \end{pmatrix} \cdot \begin{pmatrix} \overrightarrow{f^*} & \overrightarrow{-g^*} \\ -\overrightarrow{F^*} & \overrightarrow{G^*} \end{pmatrix} = \begin{pmatrix} \overrightarrow{f^*} & \overrightarrow{-g^*} \\ \overrightarrow{g} & \overrightarrow{f} \end{pmatrix}.$$

The detailed proof computations is provided in **Section A**.

At first glance, both lattices in **Table 1** are relevant candidates. Indeed, as shown in **Equation (1)**, the performance of BKZ directly depends on the ratios

$$\frac{\|(\overrightarrow{f^*}, \overrightarrow{g})\|}{\text{Vol}(\Lambda_{fg})^{1/2n}} \text{ and } \frac{\|(\overrightarrow{F^*}, \overrightarrow{G})\|}{\text{Vol}(\Lambda_{FG})^{1/2n}}.$$

It is not possible to provide an inequality between these ratios for invalidating one lattice in favor of the other. More importantly, contrarily to LWE embeddings, the volume of the lattices depend on the public key. The above ratio then varies according to the public key. However, this variation is actually artificial as the value of q_{00} and q_{11} should not impact the security of the scheme. In the next section, we show that it is actually possible to distort the lattice to obtain a unitary volume and reduce the ratio by choosing a specific matrix $\mathbf{M}$ such that $\frac{\|\mathbf{M} \cdot (\overrightarrow{f^*}, \overrightarrow{g})\|}{\text{Vol}(\mathbf{M}\Lambda)^{1/\dim(\Lambda)}} = 1$.

Table 1: Lattices and bases related to HAWK's private key. All displayed bases belong to $\mathbb{Z}^{2n \times 2n}$.

Lattice	Λ_{fg}	Λ_{FG}
Good basis	$\begin{pmatrix} \overrightarrow{f^*} & \overrightarrow{-g^*} \\ \overrightarrow{g} & \overrightarrow{f} \end{pmatrix}$	$\begin{pmatrix} \overrightarrow{F^*} & \overrightarrow{-G^*} \\ \overrightarrow{-G} & \overrightarrow{F} \end{pmatrix}$
Bad basis (public)	$\begin{pmatrix} I_n & 0_n \\ \overrightarrow{q_{01}^*} & \overrightarrow{q_{00}} \end{pmatrix}$	$\begin{pmatrix} I_n & 0_n \\ \overrightarrow{q_{01}} & \overrightarrow{q_{11}} \end{pmatrix}$

3.2 Embedding LIP into a better SVP problem with distortion

In this section, we go beyond the previous SVP embedding [CME⁺25] and introduce a distortion of Λ_{fg} of Lemma 1 that simultaneously normalizes the target vectors and sets the lattice volume to 1. It is presented in the next theorem.

Theorem 1 (Distortion). *Let $\mathbf{Q} = \mathbf{B}^* \cdot \mathbf{B} = \begin{pmatrix} q_{00} & q_{01} \\ q_{01}^* & q_{11} \end{pmatrix}$ be a HAWK public key related to the private key $\mathbf{B} = \begin{pmatrix} f & F \\ g & G \end{pmatrix}$. Let $\mathbf{Y} \in \mathbb{R}^{n \times n}$ be a symmetric matrix such that $\mathbf{Y}^2 = \begin{bmatrix} q_{00}^{-1} \end{bmatrix}$. The basis in the first line of Table 2 generates the same lattice as the basis in the second line. Besides, the lattice has the following properties:*

1. $\text{Vol}(\Lambda_{\mathbf{Q}}^{\text{dist}}) = 1$
2. The good basis is orthonormal.

The proof is based on linear algebra and is detailed in Section B.

Remark 1. It is possible to define similarly a distorted lattice from Λ_{FG} from Lemma 1 with the exact same properties. Since the volume and norm of the short vectors are the same, both lattices present the same security level against lattice reduction attacks. We therefore arbitrarily choose to focus on $\Lambda_{fg}^{\text{dist}}$.

Theorem 2. *The vectors of the good basis defined in Theorem 1 all have exactly the shortest Euclidean norm of the lattice $\Lambda_{\mathbf{Q}}^{\text{dist}}$.*

Proof. Since by Theorem 1 the good basis is orthonormal, there exists an isometry between $\Lambda_{\mathbf{Q}}^{\text{dist}}$ and $\mathbb{Z}^{2n}$ that maps each column of the good basis to $(1, 0, \dots, 0), \dots, (0, 0, \dots, 1)$. Hence, as the isometry preserves the shortest vectors, and the only shortest vectors of $\mathbb{Z}^{2n}$ are the vectors $(\pm 1, 0, \dots, 0), \dots, (0, 0, \dots, \pm 1)$, the only shortest vectors of $\Lambda_{\mathbf{Q}}^{\text{dist}}$ are the columns of the good basis up to a sign. $\square$

Corollary 1. *Let $\mathbf{Q} = \begin{pmatrix} q_{00} & q_{01} \\ q_{01}^* & q_{11} \end{pmatrix}$ be a HAWK public key and let $\mathbf{Y} \in \mathbb{R}^{n \times n}$ be a symmetric matrix such that $\mathbf{Y}^2 = \begin{bmatrix} q_{00}^{-1} \end{bmatrix}$. If one is able to solve SVP (Hard problem 1) on the lattice with basis $\begin{pmatrix} \mathbf{Y} & 0_n \\ \mathbf{Y} \begin{bmatrix} q_{01}^* \end{bmatrix} & \mathbf{Y} \begin{bmatrix} q_{00} \end{bmatrix} \end{pmatrix}$ then one can solve Hard problem 5 on $\mathbf{Q}$.*

Proof. The solution of this SVP is a column of the good basis of Table 2. Hence, one can recover a negacyclic rotation of the vector $\mathbf{Y} \cdot (\vec{f}^*, \vec{g})$ up to a sign. Since $\mathbf{Y}$ is invertible, one can recover $(\vec{f}^*, \vec{g})$ up to a sign and a negacyclic rotation. One can then recover equivalent $(\vec{F}^*, \vec{G})$ with NTRUSolve and finally build a matrix $\mathbf{B}'$ that is a solution of Hard problem 5 on $\mathbf{Q}$. $\square$

3.3 Concrete security estimation with the new approach

In the next Corollary, we provide a concrete security estimation of HAWK against lattice reduction attacks based on the embedding described in Theorem 1. The complexity is

Table 2: Good and bad bases for the distorted lattice. All displayed bases belong to $\mathbb{R}^{2n \times 2n}$.

Lattice	$\Lambda_{\mathbf{Q}}^{\text{dist}}$
Good basis	$\begin{pmatrix} \mathbf{Y} \begin{bmatrix} f^* \end{bmatrix} & \mathbf{Y} \begin{bmatrix} -g^* \end{bmatrix} \\ \mathbf{Y} \begin{bmatrix} g \end{bmatrix} & \mathbf{Y} \begin{bmatrix} f \end{bmatrix} \end{pmatrix}$
Bad basis (public)	$\begin{pmatrix} \mathbf{Y} & 0_n \\ \mathbf{Y} \begin{bmatrix} q_{01}^* \end{bmatrix} & \mathbf{Y} \begin{bmatrix} q_{00} \end{bmatrix} \end{pmatrix}$

expressed in terms of the block-size β of BKZ that is necessary to solve the SVP problem defined in [Corollary 1](#).

Corollary 2. *Let β be the minimum block-size such that*

$$\sqrt{\beta/2n} \leq \delta_\beta^{2\beta-2n-1}$$

where δ_β is the root-Hermite-Factor of BKZ- β . There exists an algorithm solving [Hard problem 5](#) whose complexity is dominated by the complexity of running BKZ- β .

Proof. Let $\Lambda_{\mathbf{Q}}^{\text{dist}}$ and $\mathbf{Y}$ be defined as in [Theorem 1](#). By [Theorem 1](#) and [Theorem 2](#), we have:

$$\frac{\lambda_1(\Lambda_{\mathbf{Q}}^{\text{dist}})}{\text{Vol}(\Lambda_{\mathbf{Q}}^{\text{dist}})^{1/2n}} = 1.$$

Hence, the necessary block-size to reduce the lattice $\Lambda_{\mathbf{Q}}^{\text{dist}}$ with BKZ is given by replacing this term into the inequality of [Section 2.2](#). If one applies BKZ- β with such a β , one recovers a shortest vector of $\Lambda_{fg}^{\text{dist}}$ and hence solves [Hard problem 5](#) by [Corollary 1](#). The complexity is hence dominated by the complexity of running BKZ- β . $\square$

Remark 2. Let us remark that the complexity of [Corollary 2](#) does not depend on η , the distribution parameter for f and g . This is different from the cryptanalysis proposed in [\[BBD⁺23b\]](#) that depends on σ_{krsec} . Indeed, in our case, the distortion allows to remove the dependency on the size information of the secret key. This independence interestingly supports our approach as [Hard problem 5](#) does not involve any size constraint on the solution. Consequently, changing the distribution of f and g would not affect the complexity predicted by our attack model. Hence, the designers of HAWK could decide to replace the current distribution with another one that requires less randomness. This would not impact the security estimates of this attack. Of course, if f or g were, for example, too sparse, other attacks could become more efficient than lattice reduction.

Computation of the public basis. To concretely build the public basis of the lattice $\Lambda_{\mathbf{Q}}^{\text{dist}}$ defined in [Theorem 1](#), the symmetric matrix $\mathbf{Y}$ such that $\mathbf{Y}^2 = \boxed{q_{00}^{-1}}$ is computed using spectral decomposition:

1. Compute q_{00}^{-1} in $\mathbb{Q}[x]/(x^n + 1)$ and convert it to its negacyclic matrix representation $\mathbf{M}$. Note that $\mathbf{M}$ is symmetric positive definite. Indeed $\mathbf{M}^T = \boxed{(q_{00}^{-1})^*} = \boxed{q_{00}^{-1}} = \mathbf{M}$ and for any non-zero polynomial $a \in R_n$, we have $\vec{a}^T \mathbf{M} \vec{a} = \vec{a}^T \boxed{q_{00}^{-1}} \vec{a} = \frac{\|\vec{a}\|^2}{q_{00}} = \frac{\|\vec{a}\|^2}{\|\vec{f}\|^2 + \|\vec{g}\|^2} > 0$.
2. Since $\mathbf{M}$ is symmetric positive definite, it admits an orthogonal diagonalization $\mathbf{M} = \mathbf{V} \text{diag}(\lambda_i) \mathbf{V}^T$ with $\lambda_i > 0$. An efficient technique consists in using a Vandermonde matrix $\mathbf{V}$ constructed from the n -th roots of -1 ;
3. Set $\mathbf{Y} := \mathbf{V} \text{diag}(\sqrt{\lambda_i}) \mathbf{V}^T$, and therefore $\mathbf{Y}^2 = \mathbf{M} = \boxed{q_{00}^{-1}}$ and $\mathbf{Y}^T = \mathbf{Y}$.

Key recovery estimates. $\Lambda_{\mathbf{Q}}^{\text{dist}}$ admits $2n$ secret vectors with the same norm, they are all negacyclic rotations. One may want to obtain a speedup with breaking the symmetry and biasing the reduction towards a single short vector. This issue has already been encountered for analysing NTRU. One solution proposed in [\[DDGR20, CDH⁺20\]](#) is to add some guesses on the secret vector to break the symmetry. We present in [Table 3](#) a comparison of block sizes for different key recovery techniques. In the first column, we report the block size β_{Gram} for reducing the Gram matrix as claimed in [\[BBD⁺23b\]](#). We ignore for now the d4f [\[Duc18\]](#) optimization that can apply independently for all of

Table 3: Estimated BKZ block sizes for key recovery (d4f is ignored here.) The second column provides the average estimated block size on 10 keys of the SVP embedding of [CME⁺25].

	$\beta_{\text{Gram}}[\text{BBD}^+23\text{b}]$	$\beta_{\Lambda_{fg}}[\text{CME}^+25]$	$\beta_{\Lambda_{\mathbf{Q}}^{\text{dist}}}$	$\beta_{\Lambda_{\mathbf{Q}}^{\text{dist}}}$ with sym break (success proba)
HAWK-256	211	228	221	218 (60%)
HAWK-512	452	474	457	454 (26%)
HAWK-1024	940	963	933	930 (9%)

our results. The third column corresponds to the estimated block sizes with the lattice reduction on $\Lambda_{\mathbf{Q}}^{\text{dist}}$ following Corollary 2. The last column shows the $\beta_{\Lambda_{\mathbf{Q}}^{\text{dist}}}$ with a symmetry break tentative including 6 guesses. We can report that the symmetry break idea is not significantly beneficial here and we will later show in Figure 1 that keeping the symmetry actually leads to better experimental results compared to the prediction. The proposed approach leads to block sizes that are similar with the one obtained with the lattice reduction on Gram matrix. Interestingly, the proposed approach seems slightly beneficial for HAWK-1024. Besides, this new technique has the advantage of being a standard lattice reduction technique that can be adapted to hints as shown in the next section.

Comparison with [CME⁺25]. As mentioned above, the embedding of LIP into a SVP problem was already discussed in [CME⁺25]. However, our distortion technique allows to remove the dependency in the algebraic norm of q_{00} (or q_{11} for the second lattice) leading to strictly better block size estimates as it places the instance in the optimal case where $\lambda_1(\Lambda_{\mathbf{Q}}^{\text{dist}})/\text{Vol}(\Lambda_{\mathbf{Q}}^{\text{dist}})^{1/2n} = 1$. For example, an average on 10 instances of the block sizes directly obtained from the SVP embedding of [CME⁺25] is given on the second column of Table 3.

4 LIP with hints

As defined in the literature for LWE and NTRU [DDGR20, MN23, DGHK23, CJC25], a hint refers to any side information available about a short vector in a given lattice Λ . In our setting, we consider the lattice introduced in Theorem 1 and aim to recover the secret short vector, which we denote for clarity

$$\mathbf{s} := (\mathbf{Y} \vec{f}^*, \mathbf{Y} \vec{g}),$$

for f, g and $\mathbf{Y}$ defined in Theorem 1.

Remark 3. Note that since the secret vector is unique up to rotation and sign (Theorem 2), all the hints should be applied to the same reference secret vector. By convention, we use $(\mathbf{Y} \vec{f}^*, \mathbf{Y} \vec{g})$ as target vector. A large amount of hints will then *de facto* break the symmetry mentioned above and allow to recover this specific vector.

Note that the considered lattice has coefficients in $\mathbb{R}$. Hence, we slightly adapt the definition of hints to our setting. With the current state of the art, only a few types of hints can be leveraged to improve the recovery of the secret vector. We list below the different types of hints that can be considered in our setting:

- A **perfect hint** is the knowledge of a pair $(\mathbf{w}, h) \in \mathbb{R}^{2n} \times \mathbb{Z}$ such that $\langle \mathbf{w}, \mathbf{s} \rangle = h$.
- A **modular hint** is the knowledge of a triple $(\mathbf{w}, h, k) \in \mathbb{R}^{2n} \times \mathbb{Z} \times \mathbb{Z}$ such that $\langle \mathbf{w}, \mathbf{s} \rangle \equiv h \pmod{k}$.

- An **approximate hint** is the knowledge of a pair $(\mathbf{w}, h) \in \mathbb{R}^{2n} \times \mathbb{Z}$ and a distribution $\mathcal{D}$ such that $\langle \mathbf{w}, \mathbf{s} \rangle = h + e$ where $e \leftarrow \mathcal{D}$.
- An **inequality hint** is the knowledge of a pair $(\mathbf{w}, h) \in \mathbb{R}^{2n} \times \mathbb{Z}$ such that $\langle \mathbf{w}, \mathbf{s} \rangle \leq h$. This hint is often considered as an approximate hint.
- A **short vector hint** is the knowledge of a lattice vector $\mathbf{w} \in \mathbb{R}^{2n}$ with a short norm.

A perfect, approximate and modular hint is said *homogeneous* when $h = 0$.

Hint conversions. One interesting property of HAWK is the fact that one can get side information on f and g but also on F and G , that are not directly related to the secret vector of $\Lambda_{\mathbf{Q}}^{\text{dist}}$. The attack surface is thus increased. As described in [Property 1](#), any information on f, g, F and G is somehow equivalent. In the following Theorem, we assume that a hint is given on either $(\vec{f}, \vec{g}), (\vec{f}, \vec{F}), (\vec{g}, \vec{G})$ and $(\vec{G}, \vec{G})$ and we show that we can derive a hint on the secret vector $\mathbf{s} = (\mathbf{Y}\vec{f}^*, \mathbf{Y}\vec{g})$ which is the shortest vector of the lattice defined by the basis $\Lambda_{\mathbf{Q}}^{\text{dist}}$ as in [Theorem 1](#).

For each case, we consider that the hint vector $\vec{w}$ for $(\vec{f}, \vec{g}), (\vec{f}, \vec{F}), (\vec{g}, \vec{G})$ or $(\vec{G}, \vec{G})$ is in $\mathbb{Z}^{2n}$. The conversion however leads to real value vectors since $\mathbf{Y}$ has real coefficients.

Theorem 3. Let $(\mathbf{w}, h) \in \mathbb{Z}^{2n} \times \mathbb{Z}$. Let us denote $\mathbf{w}$ as $\mathbf{w} = \begin{pmatrix} \vec{w}_1 \\ \vec{w}_2 \end{pmatrix}$ with $w_1, w_2 \in \mathbb{Z}^n$ being polynomials in R_n . Let f, g, F, G be secret polynomials associated with a public key $\mathbf{Q}$ and $\mathbf{Y}$ as defined in [Theorem 1](#), the following implications hold:

$$\begin{aligned}
1. \underbrace{\left\langle \mathbf{w}, \begin{pmatrix} \vec{f} \\ \vec{g} \end{pmatrix} \right\rangle = h}_{\text{hint on } (\vec{f}, \vec{g})} &\implies \underbrace{\left\langle \begin{pmatrix} \mathbf{Y}^{-1}\vec{w}_1^* \\ \mathbf{Y}^{-1}\vec{w}_2 \end{pmatrix}, \begin{pmatrix} \mathbf{Y}\vec{f}^* \\ \mathbf{Y}\vec{g} \end{pmatrix} \right\rangle = h}_{\text{hint on } \mathbf{s}=(\mathbf{Y}\vec{f}^*, \mathbf{Y}\vec{g})} \\
2. \underbrace{\left\langle \mathbf{w}, \begin{pmatrix} \vec{f} \\ \vec{F} \end{pmatrix} \right\rangle = h}_{\text{hint on } (\vec{f}, \vec{F})} &\implies \underbrace{\left\langle \begin{pmatrix} \mathbf{Y}^{-1} & \mathbf{Y}^{-1} \begin{bmatrix} q_{01}q_{00}^{-1} \\ q_{00}^{-1} \end{bmatrix} \\ 0_n & -\mathbf{Y}^{-1} \begin{bmatrix} q_{00}^{-1} \end{bmatrix} \end{pmatrix} \cdot \begin{pmatrix} \vec{w}_1^* \\ \vec{w}_2 \end{pmatrix}, \mathbf{s} \right\rangle = h}_{\text{hint on } \mathbf{s}} \\
3. \underbrace{\left\langle \mathbf{w}, \begin{pmatrix} \vec{g} \\ \vec{G} \end{pmatrix} \right\rangle = h}_{\text{hint on } (\vec{g}, \vec{G})} &\implies \underbrace{\left\langle \begin{pmatrix} 0_n & \mathbf{Y}^{-1} \begin{bmatrix} q_{00}^{-1} \end{bmatrix} \\ \mathbf{Y}^{-1} & \mathbf{Y}^{-1} \begin{bmatrix} q_{01}^* \cdot q_{00}^{-1} \end{bmatrix} \end{pmatrix} \cdot \begin{pmatrix} \vec{w}_1^* \\ \vec{w}_2 \end{pmatrix}, \mathbf{s} \right\rangle = h}_{\text{hint on } \mathbf{s}} \\
4. \underbrace{\left\langle \mathbf{w}, \begin{pmatrix} \vec{F} \\ \vec{G} \end{pmatrix} \right\rangle = h}_{\text{hint on } (\vec{F}, \vec{G})} &\implies \underbrace{\left\langle \begin{pmatrix} \mathbf{Y}^{-1} \begin{bmatrix} q_{01}q_{00}^{-1} \\ -q_{00}^{-1} \end{bmatrix} & \mathbf{Y}^{-1} \begin{bmatrix} q_{01}^2q_{00}^{-1} - q_{11} \\ -q_{01}^*q_{00}^{-1} \end{bmatrix} \\ \mathbf{Y}^{-1} \begin{bmatrix} q_{00}^{-1} \end{bmatrix} & \mathbf{Y}^{-1} \begin{bmatrix} -q_{01}^*q_{00}^{-1} \end{bmatrix} \end{pmatrix} \cdot \begin{pmatrix} \vec{w}_1^* \\ \vec{w}_2 \end{pmatrix}, \mathbf{s} \right\rangle = h}_{\text{hint on } \mathbf{s}}
\end{aligned}$$

The same reasoning holds for converting modular, approximate and inequality hints.

The detailed proofs of [Theorem 3](#) are based on linear algebra and can be found in [Section C](#).

We introduce the following corollary that gives the norm of the converted hint vectors. The proof is given in [Section C.5](#).

Corollary 3. *Let $(\mathbf{w}, h) \in \mathbb{Z}^{2n} \times \mathbb{Z}$. Let us denote $\mathbf{w}$ as $\mathbf{w} = \begin{pmatrix} \vec{w_1} \\ \vec{w_2} \end{pmatrix}$ with $w_1, w_2 \in \mathbb{Z}^n$ being polynomials in R_n . Let f, g, F, G be secret polynomials associated with a public key $\mathbf{Q}$ and $\mathbf{Y}$ as defined in [Theorem 1](#).*

$$\text{Let } \mathbf{v}_1 = \begin{pmatrix} \mathbf{Y}^{-1} & \mathbf{Y}^{-1} \boxed{q_{01} q_{00}^{-1}} \\ 0_n & -\mathbf{Y}^{-1} \boxed{q_{00}^{-1}} \end{pmatrix} \cdot \begin{pmatrix} \vec{w_1}^* \\ \vec{w_2}^* \end{pmatrix}, \text{ and let } \mathbf{v}_2 = \begin{pmatrix} 0_n & \mathbf{Y}^{-1} \boxed{q_{00}^{-1}} \\ \mathbf{Y}^{-1} & \mathbf{Y}^{-1} \boxed{q_{01}^* \cdot q_{00}^{-1}} \end{pmatrix} \begin{pmatrix} \vec{w_1} \\ \vec{w_2} \end{pmatrix},$$

then $\|\mathbf{v}_1\|^2 = \|\mathbf{v}_2\|^2 = \|\mathbf{w}\|_Q^2$.

5 Integration in LeakyLWEEstimator

To validate this work, we developped two types of scripts on SageMath: a prediction script for estimating the block size on high dimensions and an attack script for concretely applying the attack on small dimensions. The prediction script is based on the BKZ simulator of [\[ACD⁺18, DDGR20\]](#) and is used to estimate the security of large lattices without using the full [\[DDGR20\]](#) framework, it only computes the block size as a function of the lattice parameters (but not using the actual basis). The attack script is used to validate the integration of hints and the accuracy of the predictions. It consists in a class `instance_LIP.sage` that allows to embed a LIP instance into a basis for the LeakyLWEEstimator tool [\[DSDG⁺22\]²](#). In this section, we present some details on the attack script.

Initialization of the instance, precision and conversions. The instance is created with a lattice basis Λ_Q^{dist} as defined in [Theorem 1](#). For the covariance matrix, the distribution of the shortest vector is the distribution of a vector $\mathbf{s} \in \Lambda_Q^{\text{dist}}$ such that $\|\mathbf{s}\|_2 = 1$. Thus, we initialize the covariance matrix with $(1/2n) \cdot \mathbf{I}_{2n}$. The lattice basis is not integer, as it was in the original tool. However, the framework already natively supported rational numbers as the covariance matrix is not required to be integer. The basis of the lattice is thus first approximated to a rational matrix with enough precision. The precision is a parameter of the class and can be adapted. We report that a precision of 50 bits is enough to get stable results for the estimation and attacks in our parameter sets. In complement, we created a class `conversions.sage` that applies all the results from [Theorem 3](#). The formulas have been validated experimentally on many examples.

Validation of the key recovery. HAWK does not have small instances for testing cryptanalysis. Hence, we created small versions for $n = 16, 32$ and 64 in order to check if the recovered shortest vector corresponds to a negacyclic rotation of the secret key. We denote these instances as Small-HAWK. We report recovering a negacyclic rotation of the secret key in 100% of our tests for small instances. Over 1000 tests, the average BKZ block size for which the attack succeeds is 2 for Small-HAWK-16 and Small-HAWK-32, and 30 for Small-HAWK-64.

Homogeneity. To integrate hints, the lattice should contain the possibility of embedding the inhomogeneous part of the hints i.e. when $h \neq 0$. For LWE lattices, Kannan embedding [\[BSW16\]](#) allows to add a fixed value, often fixed as 1, to the secret vector. Hence, this fixed value can be used to embed the inhomogeneous part of the hints. For the LIP lattice, no embedding is used and the lattice can be seen as homogeneous as no extra part allows to

²We initially planned to integrate these scripts into the public LeakyLWEEstimator repository. However, in light of recent attacks [\[SW26, MPM26\]](#), we postpone this integration until the long-term relevance of studying LIP instances and supporting them in the tool is clearer.

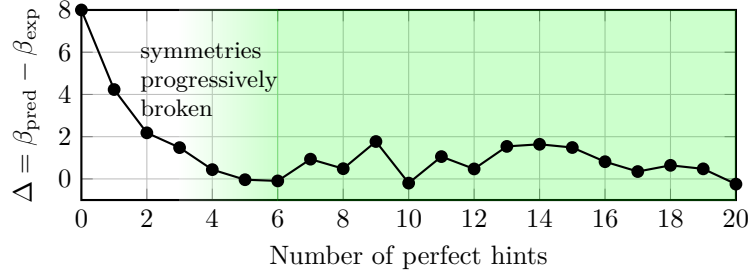


Figure 1: Evolution of the average over 100 tests of the difference $\beta_{\text{pred}} - \beta_{\text{exp}}$ as a function of the number of perfect hints for $n = 64$.

embed the inhomogeneous part. To integrate inhomogeneous hints, an adaptation must be made. Working in a lattice coset is actually not trivial: finding a coset shift that belongs to $\Lambda_{\mathbf{Q}}^{\text{dist}}$ and that is small is not easy and may involve the shortest vector itself.

We address the issue in two ways.

The first solution is to use the first perfect inhomogeneous hint as an embedding for the other inhomogeneous hints. More precisely, let $\mathbf{w}$ be a hint vector and $h \neq 0$ be its hint value. This hint is "sacrificed" to embed the inhomogeneous parts of the next inhomogeneous hints. For any other hint $\mathbf{w}'$ with hint value $h' \neq 0$, the hint $\mathbf{w}' - h'/h \cdot \mathbf{w}$ is the corresponding homogeneous hint. Indeed,

$$\left\langle \mathbf{w}' - \frac{h'}{h} \cdot \mathbf{w}, \mathbf{s} \right\rangle = \langle \mathbf{w}', \mathbf{s} \rangle - \frac{h'}{h} \cdot \langle \mathbf{w}, \mathbf{s} \rangle = h' - h' = 0.$$

Alternatively, it is also possible to make a mild guess: for example assume that the first coordinate of the secret key is 1. That way, this first coefficient allows to embed the non-homogeneous part for all hints. In other words, such a guess implies that there exists a vector $\mathbf{v}$ that can be deduced from $\mathbf{Y}$ such that $\langle \mathbf{v}, \mathbf{s} \rangle = 1$. Hence, let $\mathbf{w}$ be a hint vector and h be its hint value, the hint $\mathbf{w} - h \cdot \mathbf{v}$ is the corresponding homogeneous hint. Making such a guess implies a small constant factor of attack restart. If the first coefficient is zero, one can guess that the second coefficient is 1 and apply the same strategy, and so on.

Validating hint integration and predictions. For validating the hint integration, we used Small-HAWK-64 and we generated 20 random perfect, modular and approximate hints. We report recovering a rotation of the secret key in 100% of our tests for Small-HAWK-64. To refine the accuracy of the predictions in this regime of small block sizes, we use the probabilistic simulation method of [DDGR20, Figure 4]. In Figure 1, we report the evolution of the difference $\beta_{\text{pred}} - \beta_{\text{exp}}$ as a function of the number of perfect hints with 100 tests (similar results can be obtained with modular and approximate hints and are omitted for brevity). We observe an interesting phenomenon: the predictions are overestimated when the number of hints is below 4, which is probably due to the negacyclic structure (symmetries) of the lattice. For a larger number of hints, the symmetries are broken and the predictions become more accurate. Such a phenomenon is also observed for modular and approximate hints, but the threshold happens later (for example with around 17 modular hints with modulus 2), which is consistent with the fact that modular and approximate hints are less constraining than perfect hints, hence breaking the symmetries more slowly. We also report that the predictions are as accurate as in [DDGR20] for a large number of hints, which validates the hint integration for LIP.

Too many hints regime. When the number of hints is large enough, the attack can be simplified by directly recovering the secret key from the hints [MN23]. The proposed

methodology is completely compatible with this regime. However, the threshold of the "too many hints" regime depends on the parameters and on the distribution of the hints. Estimating such a threshold is left as an interesting future work. If the behaviour is close to other LWE-based schemes, this threshold is around n perfect hints.

6 Applications

6.1 Hints given from signatures

Now that the embedding of LIP into SVP is compatible with hints, the next step is to determine whether the scheme itself provides intrinsic hints. Unlike LWE-based schemes, signatures in HAWK can reveal information about the secret key. This work shows that *overly short signatures* and *overly large signatures* both threaten the security of the scheme. Exceptionally short signatures yield useful short-vector hints, while unusually large signatures yield strong a-posteriori approximate hints on the secret key. All results are obtained with our `prediction` script.

Short vector hints from signatures. The security of HAWK is based on the "one more SVP" problem, a variant of SVP in which the attacker has access to several short lattice vectors. In particular, we show here that the vector $\mathbf{w}$, that can be extracted from the signatures, can be used to generate short vectors in the lattice $\Lambda_{\mathbf{Q}}^{\text{dist}}$.

Let $\mathbf{w} = \begin{pmatrix} w_0 \\ w_1 \end{pmatrix}$ be obtained from a signature (we refer to [Section 2.4](#) for details).

The vector $\mathbf{w}' := \begin{pmatrix} \mathbf{Y} & 0_n \\ \mathbf{Y}_{\boxed{q_{01}}}^* & \mathbf{Y}_{\boxed{q_{00}}} \end{pmatrix} \cdot \mathbf{w}$ is a vector of the lattice $\Lambda_{\mathbf{Q}}^{\text{dist}}$, and its norm is $\|\mathbf{w}'\|^2 = \|\mathbf{w}\|_{\mathbf{Q}}^2 = \|\mathbf{x}\|^2$ where $\mathbf{x}$ is the corresponding small sample. All negacyclic rotations of $\mathbf{w}'$ are also vectors of the lattice, and they all have the same norm $\|\mathbf{w}\|_{\mathbf{Q}}^2$. Hence, each signature provides $2n$ short vector hints of norm $\|\mathbf{w}\|_{\mathbf{Q}}$. In the sequel, we assume that all the negacyclic rotations of $\mathbf{w}$ are linearly independent, which happens with overwhelming probability. The average norm of these hints is $\sqrt{2n}(2\sigma_{\text{sign}})$.

According to [\[DDGR20\]](#), such short-vector hints can reduce the BKZ block size by projecting the lattice orthogonally to the hint vector. The final short vector recovered after projection does not directly reveal the target secret, since the projection is not injective. Nevertheless, as long as the number k of integrated short vectors remains small compared to n , the attack is still expected to succeed, as this is essentially equivalent to having $2n - k$ perfect hints. A cost of an additional mild lattice-reduction step is expected, but this does not change the asymptotic cost of the attack.

To predict the block size, according to [\[DDGR20, Section 4.5 and Fact 14\]](#), the effect of a single short-vector hint is paradoxical: the dimension decreases by 1 (which helps the attack), while the volume is divided by the norm of the hint (which makes the attack harder when the norm is large). Concretely, integrating a single short-vector hint amounts to finding the minimum β such that

$$\sqrt{\beta/(2n-1)} \leq \left(\frac{1}{\|\mathbf{w}\|_{\mathbf{Q}}} \right)^{\frac{1}{2n-1}} \delta_{\beta}^{2\beta-2n-2}.$$

In [Figure 2](#), we report the predicted block size β as a function of the hint norm $\|\mathbf{w}\|_{\mathbf{Q}}$, expressed as a percentage of its average norm. Since one signature provides at most $2n$ short-vector hints, our script integrates short-vector hints until doing so is no longer beneficial, (this always happens with $k \leq n$ short vector hints). We see that this integration is beneficial only when the signature norm is below 20% of the average norm.

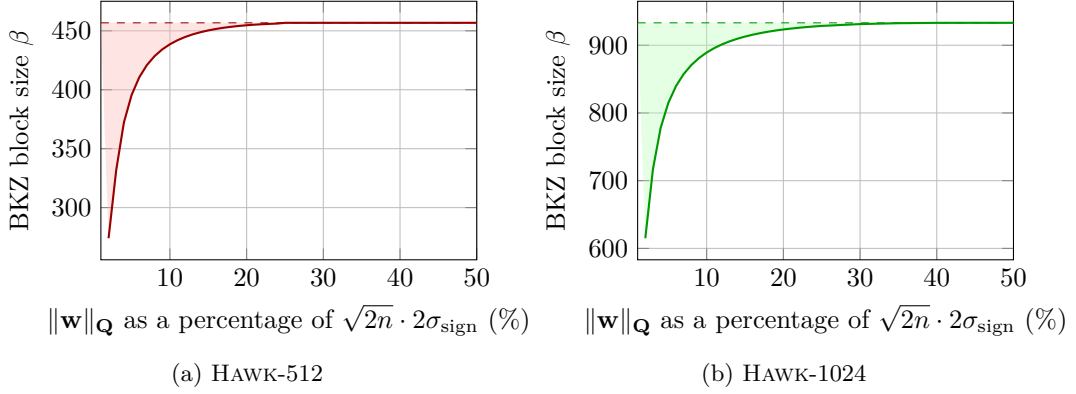


Figure 2: Predicted BKZ block size β after integrating short vector hints given by a signature, as a function of the percentage of the average norm of $\|\mathbf{w}\|_{\mathbf{Q}}$, given by $\sqrt{2n} \cdot 2\sigma_{\text{sign}}$. The colored area represents the range of short vector hint norms for which the integration is beneficial. Similar results can be obtained for HAWK-256 but are omitted to save space.

Using tail-bound computations, this event has negligible probability for HAWK parameters. Hence, although signatures do provide short-vector hints, *they are not short enough to affect security with our current technique.*

However, if the Gaussian sampler can be faulted so as to produce smaller signatures, then an interesting attack path is possible. For example, if one is able to generate a faulted signature with a norm below 1% of the average norm, the security drops to around $274 \times 0.292 = 80$ (resp. $98 \times 0.292 = 29$) bits of security for HAWK-512 (resp. HAWK-256). Such a fault model could require to draw valid signatures by faulting the Gaussian sampler to output more zeros than expected without targeting any specific iteration. Determining whether such a fault model is realistic is left as an interesting future direction.

A-posteriori approximate hints from signatures. Let us now consider the opposite extreme, namely large signatures. In this case, signatures can be used as a-posteriori approximate hints on the secret key. More precisely, each signature yields a collection of n a-posteriori approximate hints on $(f - F)$ and n a-posteriori approximate hints on $(g - G)$. Indeed, during signature generation, a sample $\mathbf{x}$ is drawn such that $\mathbf{x} = \mathbf{B} \cdot \mathbf{w}$ with a known $\mathbf{w}$. The value of $\mathbf{x}$ is unknown, but its distribution is known. As mentioned in Section 2.4, this distribution is not exactly a discrete Gaussian. However, its mean and standard deviation are easy to derive: it is centered at 0 and has standard deviation $2\sigma_{\text{sign}}$, which is a public parameter. Hence, each signature provides a family of $2n$ a-posteriori approximate hints on the secret key. We recall that we assume that all hints are linearly independent, which happens with overwhelming probability. The a-posteriori nature of these hints means that the additional knowledge is not conditioning the distribution of the secret short vector, but rather replacing it. Hence, only one signature can be used to decrease the BKZ block size. The gain obtained by integrating one such signature depends on the norm of its corresponding $\mathbf{w}$. To obtain a concrete estimate, we refine [DDGR20] and derive a prediction as a function of $\|\mathbf{w}\|$.

Lemma 2. *Let $\Lambda_{\mathbf{Q}}^{\text{dist}}$ be a lattice derived from a HAWK public key and its corresponding short vector $\mathbf{s}$. Let $\mathbf{v}$ be a homogeneous a-posteriori approximate hint with standard deviation σ , that is, $\mathbf{v}$ is known and satisfies*

$$\mathbb{E}[\langle \mathbf{v} | \mathbf{s} \rangle] = 0, \quad \text{Var}[\langle \mathbf{v} | \mathbf{s} \rangle] = \sigma^2.$$

Then the new a-posteriori BKZ block size β after integrating the hint is the minimum β such that

$$\sqrt{\beta/2n} \leq \left(\frac{\|\mathbf{v}\|}{\sqrt{2n}\sigma} \right)^{1/2n} \delta_\beta^{2\beta-2n-1}.$$

Proof. Integrating the information carried by the a-posteriori hint first amounts to modifying the covariance matrix as follows (see [DDGR20, section 4.4]):

$$\Sigma' = \Sigma + \left(\frac{\sigma^2}{(\mathbf{v}^T \Sigma \mathbf{v})^2} - \frac{1}{\mathbf{v}^T \Sigma \mathbf{v}} \right) \Sigma \mathbf{v} (\Sigma \mathbf{v})^T.$$

We can rewrite Σ' as $\Sigma' = \Sigma + \alpha \mathbf{U} \mathbf{U}^T$, where $\alpha = \frac{\sigma^2}{(\mathbf{v}^T \Sigma \mathbf{v})^2} - \frac{1}{\mathbf{v}^T \Sigma \mathbf{v}}$ and $\mathbf{U} = \Sigma \mathbf{v}$. By applying the matrix determinant lemma, we get $\det(\Sigma') = \det(\Sigma) \cdot (1 + \alpha \mathbf{U}^T \Sigma^{-1} \mathbf{U})$. We now compute the scalar term:

$$\begin{aligned} 1 + \alpha \mathbf{U}^T \Sigma^{-1} \mathbf{U} &= 1 + \left(\frac{\sigma^2}{(\mathbf{v}^T \Sigma \mathbf{v})^2} - \frac{1}{\mathbf{v}^T \Sigma \mathbf{v}} \right) \mathbf{v}^T \Sigma \Sigma^{-1} \Sigma \mathbf{v} \\ &= 1 + \frac{\sigma^2}{\mathbf{v}^T \Sigma \mathbf{v}} - 1 = \frac{\sigma^2}{\mathbf{v}^T \Sigma \mathbf{v}}. \end{aligned}$$

Hence, $\det(\Sigma') = \det(\Sigma) \cdot \frac{\sigma^2}{\mathbf{v}^T \Sigma \mathbf{v}}$. Secondly, to incorporate the a-posteriori information into the lattice, the basis is multiplied by $\Sigma'^{-1/2}$, which multiplies the volume by $\det(\Sigma')^{-1/2}$. Therefore, the volume is multiplied by $(\sqrt{2n})^{2n} \sqrt{\frac{\mathbf{v}^T \Sigma \mathbf{v}}{\sigma^2}} = (\sqrt{2n})^{2n} \sqrt{\frac{\|\mathbf{v}\|^2}{2n\sigma^2}}$. This also yields an instance whose target covariance is the identity matrix, so $\|\mathbf{s}\|$ is replaced by $\sqrt{2n}$ in Equation (1). Substituting these terms into Equation (1) gives the new prediction for the required BKZ block size β . $\square$

Recall that the vector $\mathbf{w}$ provides hints on $(f \ F)$ and $(g \ G)$. A conversion using Theorem 3 is needed to obtain the corresponding $\mathbf{v}$ required in Lemma 2. Using Corollary 3, the residual BKZ block size β after integrating all $2n$ pieces of a-posteriori information provided by a signature is the minimum β such that

$$\sqrt{\beta/2n} \leq \frac{\|\mathbf{w}\|_{\mathbf{Q}}}{\sqrt{2n}2\sigma_{\text{sign}}} \delta_\beta^{2\beta-2n-1}. \quad (3)$$

Since the norm $\|\mathbf{w}\|_{\mathbf{Q}}$ coincides with $\|\mathbf{x}\|$, it satisfies

$$\mathbb{E}(\|\mathbf{w}\|_{\mathbf{Q}}^2) = \sqrt{2n}2\sigma_{\text{sign}} \text{ and } \|\mathbf{w}\|_{\mathbf{Q}} < \sqrt{2n}2\sigma_{\text{ver}},$$

where $\sigma_{\text{ver}} > \sigma_{\text{sign}}$ is chosen as a parameter. Substituting the expected value into Equation (3) yields the same equation as in Corollary 2, which means that *an average signature does not reduce the BKZ block size*. The natural strategy is therefore to look for signatures such that $\|\mathbf{w}\|_{\mathbf{Q}}$ is close to $\sqrt{2n}2\sigma_{\text{ver}}$. In Figure 3, we present the predicted BKZ block size β as a function of a percentage of $\sqrt{2n}2\sigma_{\text{ver}}$. One can see that the BKZ block size decreases significantly when the signature norm is close to the upper bound.

It is therefore natural to look for a signature whose norm is close to this bound. This can be done in a first phase. The restart probability of the norm check can be estimated with the function `failProbability` from [PDPvW23]. For $n = 256$ (resp. $n = 512$), one can obtain signatures arbitrarily close to the bound, as the estimate predicts a cost of about 2^3 (resp. 2^{23}) signatures. However, for $n = 1024$, reaching the bound is essentially negligible, as it would require around 2^{128} signatures. Hence, with a budget limited to 2^{64} signatures, the estimate of [PDPvW23] predicts a norm of $0.94 \cdot \sqrt{2n}(2\sigma_{\text{ver}})$ with an expected amount of 2^{56} signatures. We report in Table 4 the resulting a-posteriori BKZ block size β under

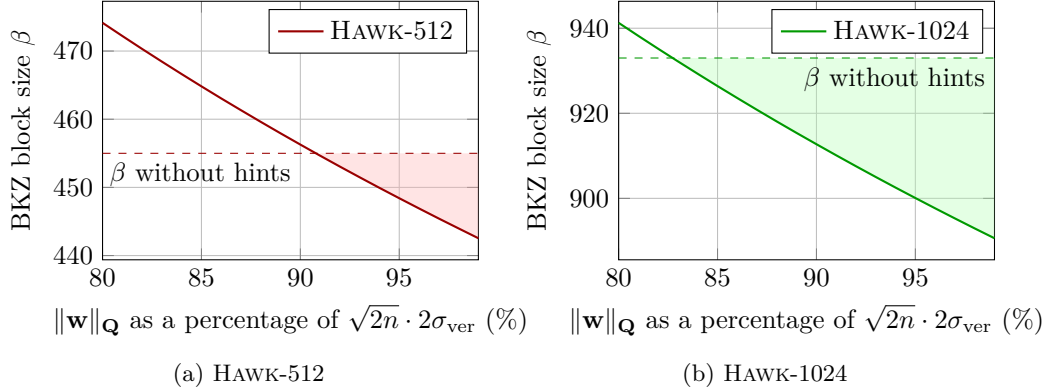


Figure 3: Predicted BKZ block size β after integrating all $2n$ a-posteriori approximate hints given by a signature, as a function of the percentage of the maximum norm of $\mathbf{w}$. The colored area represents the range of hint norms for which the integration is beneficial. Similar results can be obtained for HAWK-256 but are omitted to save space.

Table 4: Refined BKZ block size estimates for key recovery using the a-posteriori information provided by signatures

	[BBD ⁺ 23b]		This work, leveraging at most 2^{64} signatures		
	β	β (d4f)	β	β (d4f)	bit security w.r.t. [ACD ⁺ 18]
HAWK-256	211	187	218	193	56
HAWK-512	452	412	441	402	117
HAWK-1024	940	873	902	837	244

this budget. We also add a column that includes the dimensions for free (d4f) optimization from [Duc18] to provide a final comparison and an estimated bit security in [ACD⁺18] cost model. The gain is relatively significant for HAWK-512 and HAWK-1024, and eats into the designer’s security margin. For example, HAWK-1024 loses about $(873 - 837) \times 0.292 \approx 11$ bits of security, dropping to an estimated 244 bits of security. This security loss can still be considered as inside the security margin. Indeed, the estimates of ML-DSA consider a blocksize of 818 still acceptable for NIST level IV [LDK⁺22, Table 4]. We still emphasize this estimate uses only public signature information and does not rely on any side-channel measurement.

Note that our approach is more beneficial for high security parameters, particularly for HAWK-1024. This is due to the larger value of σ_{ver} that allows to reach larger signatures with a budget of 2^{64} signatures, while for HAWK-256 and HAWK-512, the budget of 2^{64} would allow to reach larger signatures but they are rejected by the bound of $\sqrt{2n}2\sigma_{\text{ver}}$.

Here again, a fault attack could be used to force the signature norm close to the bound, or even to increase the bound itself, which would significantly decrease the security of the scheme.

In conclusion, the parameters σ_{sign} and σ_{ver} , which are initially chosen to ensure correctness and to protect against signature forgery, also have a significant impact on key-recovery security. Choosing σ_{sign} too small or σ_{ver} too large can therefore lead to substantial security losses.

6.2 An alternative Property 1

In [Property 1](#), the proposed algorithm is polynomial but quantum. However, it is also possible to recover the secret key with a classical algorithm by using the hints to decrease the dimension of the lattice and then applying a lattice reduction attack. The resulting attack could be considered as a more practical alternative to the quantum algorithm. The knowledge of one of the four polynomials f, g, F, G corresponds to n perfect hints. Hence, one can easily deduce the required block size, as reported in [Table 5](#). The security parameter is roughly halved compared to the one of the original attack.

Remark 4. The high number of hints (i.e. n) may translate the attack into a "too many hints" regime, where the attack is not more efficient than a direct attack without hints. However, as the hints are in canonical directions, the hint matrix will not provide any information about the other part of the secret key. Hence, in this particular setting, the estimated block size should necessarily be on the lattice $\Lambda_{\mathbf{Q}}^{\text{dist}}$.

Table 5: Estimated BKZ block sizes for key recovery when one of the four polynomials f, g, F, G is known (d4f is ignored here.)

	$\beta_{\Lambda_{\mathbf{Q}}^{\text{dist}}}$
HAWK-256	104
HAWK-512	220
HAWK-1024	457

6.3 Application to side-channel analysis of HAWK

In [\[GR24\]](#), the authors present a side-channel attack on the HAWK signature scheme. The attack relies on the recovery of some information about the coefficients of $\mathbf{x}$. The recovered information is either the sign³ or the parity of some coefficients or even the exact value of some coefficients. The device allowed to recover almost all the secret coefficients, and the authors could perform a slight brute force. However, if the measurements would not allow to recover the key by brute force, the authors left the question of how to integrate such information in the estimation of the residual complexity of the attack open. We present in this section how to integrate such information in the estimation of the residual complexity of the attack and we give an estimation of the gain obtained with such information.

Parity knowledge. Let us start by analyzing the parity knowledge. One can see the equation $\mathbf{x} = \mathbf{B} \cdot \mathbf{w} \bmod 2$ a collection of n modular hints on $(f - F)$ and n modular hints on $(g - G)$. Using our [Theorem 3](#), we can convert these modular hints into modular hints on the secret $\mathbf{s}$ of the lattice $\Lambda_{\mathbf{Q}}^{\text{dist}}$, derived from $\mathbf{Q}$ the public key. We directly use our `prediction` script to compute directly the new volume of the lattice after the integration of the modular hints by multiplying it by 2^{2n} and then to compute the new BKZ block size with the new volume. We recall that we assume that all hints are linearly independent, which happens with overwhelming probability. We report in the second column of [Table 7](#) the estimated residual β after integrating the parity information of $\mathbf{x}$.

If there is uncertainty about the parity, one can simply filter the coefficients of $\mathbf{x}$ for which the parity is not clear and only integrate the information for the coefficients for

³Note that the sign should be known after the symbreak step [\[BBD⁺23b, Step 15 of Algorithm 15\]](#).

Table 6: Type of side-channel information

	Gained knowledge	Type of hint
parity	$\mathbf{x}[i] \bmod 2 = 0 \text{ or } 1$	Modular hint
sign	$\mathbf{x}[i] \geq 0 \text{ or } \leq 0$	Inequality hint or approximate hint
value	value of $\mathbf{x}[i]$	Perfect hint

which the parity is known with high confidence. In that case, the number of required signature traces can be increased depending on the confidence of the parity information.

Sign knowledge. Moving to the sign information, it can be modeled as a collection of n inequality hints on $(f - F)$ and n inequality hints on $(g - G)$. Similarly to Section 6.1, we use Theorem 3 and Lemma 2 to estimate the gain obtained with the integration of such inequality hints. The residual BKZ block size β after integrating $2n$ a-posteriori information given by the knowledge of the sign of the coefficients of $\mathbf{x}$ is the minimum β such that

$$\sqrt{\beta/2n} \leq \frac{\|\mathbf{w}\|_{\mathbf{Q}}}{\sqrt{2n}\sigma_{\text{aposteriori}}} \delta_{\beta}^{2\beta-2n-1},$$

where $\sigma_{\text{aposteriori}}$ is the standard deviation of the distribution of the coefficients of $\mathbf{x}$ conditioned to the knowledge of the sign. We report in the third column of Table 7 the estimated residual β after integrating the sign information for $\|\mathbf{w}\|_{\mathbf{Q}} = \sqrt{2n}\sigma_{\text{sign}}$, the average norm. We can see that the gain obtained with the sign information is significant, but less than the gain obtained with the parity information. This is due to the fact that the sign information is less constraining than the parity information.

Remark 5. It is also possible to use [DGHK23], a modification of [DDGR20] that allows an estimation of inequality hints. The modification of the volume corresponds to the equations presented in [DGHK23, Section 4.1] with $\alpha = 0$ (as the inequality is comparing to zero) and $r = d$. This geometric estimation technique provides in theory more insurance on the final estimation but as a consequence the recovered block size in our case are only slightly affected by the integration of the hints. This approach is more relevant when α is close to 1.

One can also remark that the gains from parity and sign information can be cumulated. Indeed, the sparsification and distortion of the lattice are compatible and can be integrated together in the estimation of the residual complexity of the attack. We report in the last column of Table 7 the estimated residual β after integrating both the parity and sign information. The security parameter is roughly divided by 2 for Hawk-256 and divided by 1.5 for Hawk-512 and Hawk-1024.

Exact value knowledge. To finish, the exact knowledge of a coefficient of $\mathbf{x}$ can be integrated as a perfect hint on the corresponding coefficient of $\mathbf{s}$. The exact value knowledge is the most powerful type of information. To predict the average gain obtained, one can compute directly the new block size by subtracting the number of hints from the dimension of the lattice and computing the new volume by multiplying it by the average norm of $\mathbf{w}$, given by $2\sigma_{\text{sign}}\sqrt{2n}$, and then to compute the new block size. One can see the block size progressive decrease in Figure 4. Note that the "too many hints" threshold can be reached when the number of hints is large enough. This estimation is considering that no information is given on the parity and sign, but it is also possible to cumulate the gains when hints are linearly independent.

Table 7: Estimated residual β obtained with the knowledge of the parity and/or sign of the coefficients of $\mathbf{x}$.

BKZ block size β	No information	Parity of $\mathbf{x}$	Sign of $\mathbf{x}$	Parity and sign of $\mathbf{x}$
HAWK-256	221	167	173	133
HAWK-512	457	369	383	314
HAWK-1024	933	785	808	689

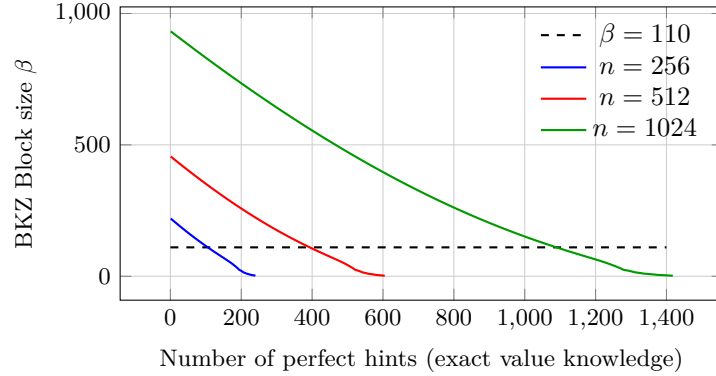


Figure 4: BKZ block size β as a function of the number of perfect hints.

Comparison with [GR24]. In the low-noise ChipWhisperer Lite / STM32F3 (Cortex M4) setting considered in [GR24], the authors recover more than 98% of the coefficients of $\mathbf{x}$ from a single trace. Combined with a mild brute-force step, this is sufficient to recover the secret key from the linear relation $\mathbf{x} = \mathbf{B} \cdot \mathbf{w}$ using two distinct pairs $(\mathbf{x}, \mathbf{w})$. Our approach can also lead to key recovery from a single trace, but it remains applicable when the side-channel information is incomplete or noisy. Therefore, in this ideal regime, the number of traces is not the most meaningful comparison metric. To better highlight the benefit of our hint-based analysis, we will consider three more realistic scenarios, each represented in a column of Table 8. For the comparisons, we cap the lattice-reduction effort available to the side-channel adversary at $\beta = 110$, representing a well-resourced attacker. This value should be understood as an illustrative threshold rather than as a precise maximum of core-hours since the concrete cost of lattice reduction depends strongly on the algorithm, implementation, hardware and degree of parallelism.⁴

Table 8: Comparison of the number of traces required to recover the secret key with a fixed computation budget corresponding to $\beta = 110$ with both approaches for Hawk-256.

Number of traces	First scenario: Half of the zero-coeffs.	Second scenario: Parity, sign, and half of the zero-coeffs.	Third scenario: All zeros with small noise.
[GR24]	28	28	N.A.
Our approach	8	4	7

In the first scenario, we keep the same experimental setting but assume that only a fraction of the intermediate values can be recovered exactly. This is realistic in view of [GR24], where some coefficients appear easier to extract than others. More precisely, we assume that the attacker recovers exactly half of the zero coefficients of $\mathbf{x}$. As explained

⁴To illustrate this dependence, extrapolating the empirical HashSieve model of [MLB15] to dimension 110 gives approximately $5.4\text{--}5.7 \times 10^3$ core-hours on the authors’ 16-core platform. In contrast, [AR24] reports wall-clock times between 1.4 and 2.5 hours for a full sieve in dimension 110, depending on the parallel configuration. More recent optimized sieving implementations further illustrate how strongly practical costs depend on the algorithm and implementation choices: [ZDY25] reports substantially improved practical performance for higher-dimensional SVP instances, including solving the dimension-179 and dimension-183 Darmstadt SVP challenges in 11.2 and 30 days, respectively, on a 112-core CPU server. These figures concern individual SVP/sieving computations and are not directly comparable to the total cost of a complete lattice reduction with a given block size. Regarding memory, [MLB15] used a machine equipped with 128 GB of RAM but did not reach dimension 110, while the dimension-110 experiments of [AR24] used nodes equipped with 192 GiB of RAM without reporting their actual peak memory consumption.

in [GR24, p. 9], these coefficients can be identified exactly by observing the sign in two different parts of the trace, since zeros may be classified as either positive or negative. This corresponds to recovering about 7% of the coefficients of $\mathbf{x}$ per trace for all HAWK parameter sets⁵.

- For [GR24]: the brute-force completion step is no longer feasible with only 7% of the coefficients known per trace. Nevertheless, one may still accumulate information from multiple traces and build a linear system involving coefficients from different vectors $\mathbf{x}$, until reaching roughly 98% of the $2 \times 2 \times n$ coefficients needed to recover the secret key are known. This leads to an estimated requirement of about $2 \times 98/7 = 28$ traces for all HAWK parameter sets.
- For our approach: according to our results in Figure 4, and for a BKZ block size $\beta = 110$, it is enough to recover exactly 55% of the coefficients of $\mathbf{x}$ to obtain a key recovery. This corresponds to about $55/7 \approx 8$ traces for all HAWK parameter sets.

In Figure 5, we plot the generalized number of traces required for a complete key recovery as a function of an arbitrary percentage of coefficients recovered per trace, for both approaches.

In a second scenario, we assume that in addition, the parity and the sign of the coefficients of $\mathbf{x}$ are also recovered, the attack of [GR24] is unchanged. In contrast, by combining the estimates from Table 7 and Figure 4, it remains to recover exactly only 25% of the coefficients, so our approach reduces the requirement to only 4 traces for all HAWK parameter sets.

In a third scenario, we now turn to a more challenging setting in which each coefficient is not recovered exactly; instead, the attacker observes the Hamming weight $HW(x)$ corrupted by a small Gaussian noise of parameter 0.33. Since the key recovery of [GR24] does not tolerate errors, it no longer applies. Our approach, however, still does. To express the result in terms of traces under the same fixed computational budget, we use the following strategy: we discard all measurements that are not equal to 0, and we collect enough traces to obtain $2n$ such measurements. This amounts to roughly $100/14 \approx 7$ traces (as there are 14% of the coefficients of $\mathbf{x}$ that are zero). Feeding these $2n$ approximate a posteriori hints into our tool yields a residual BKZ block size of $\beta = 110$ (see Figure 6). Our script is available in supplementary material.

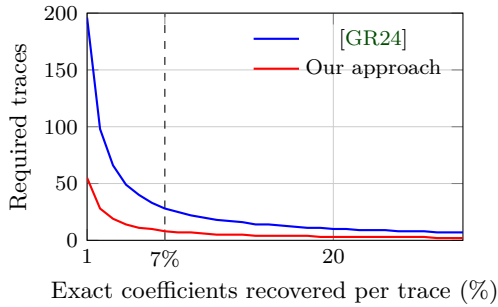


Figure 5: Number of traces required for a complete key recovery with a fixed computational budget corresponding to $\beta = 110$, identical for Hawk-256, Hawk-512, and Hawk-1024.

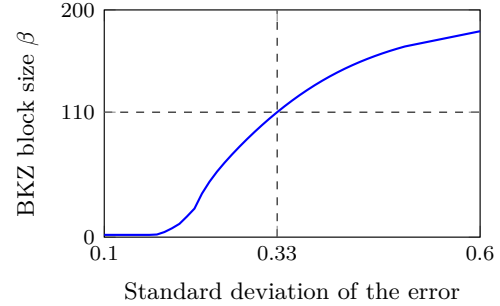


Figure 6: BKZ block size β after integration of $2n$ a posteriori approximate hints as a function of the standard deviation for Hawk-256.

These three scenarios are examples to outline the fact that our approach is more efficient and versatile than [GR24] that requires recovering exactly most of the coefficients of $\mathbf{x}$.

⁵The probability of having redundant information between traces is negligible as it requires to have signatures with the same $\mathbf{w}$ (or rotations).

7 Conclusion and future work

To conclude, this work provides a new methodology to integrate hints in the security analysis of LIP leading to insights on the black-box security of HAWK and tools for side-channel analysis.

Recent cryptanalysis of Hawk. Recent cryptanalytic results on HAWK, which appeared after the initial version of this work [SW26, MPM26], suggest a natural direction for future research: integrating the hint-based techniques developed here into the lower-dimensional lattices introduced by these attacks. Although these results already lower the estimated cost of key recovery, combining them with hints obtained from additional signatures could provide a sharper assessment of HAWK's security.

Towards a better understanding of LIP and the "one more SVP" problem. As depicted in Figure 1, the predictions of LIP without hints tend to be overestimated compared to the actual attack complexity. This is probably due to the many symmetries of the lattice, which are not taken into account in the estimation. A better understanding of the impact of symmetries on the complexity of lattice reduction algorithms would be an interesting direction for future work, and could lead to more accurate predictions for LIP and other problems with many symmetries like NTRU. Besides, as shown in Section 6.1, the signatures of HAWK are a good source of short vector hints. However, the norm of the obtained short vector hints is slightly too large to have an impact on the current predictions. Such vectors are still short compared to a bad basis of the lattice, and an interesting fundamental question is to understand how to integrate moderately short vector hints in the estimation of the complexity of lattice reduction algorithms. One idea would be to take the quality of the basis as a parameter of the estimation, it currently has no asymptotic impact but if many short vectors are available, the impact might be significant.

The open problem of non-linear hints. The proposed methodology is compatible with linear hints, but it is an open problem to include non-linear information on the shortest vector in lattice instances. This would be particularly relevant for HAWK, as the equations that relate f, g, F, G are not linear.

Linear Algebra Computations

A Details for Lemma 1

Let us look at the first case. To prove that both basis generate the same lattice, we exhibit an invertible matrix vector $\mathbf{X} \in \mathcal{GL}_{2n}(\mathbb{Z})$ such that:

$$\begin{pmatrix} I_n & 0_n \\ \boxed{q_{01}^*} & \boxed{q_{00}} \end{pmatrix} \cdot \mathbf{X} = \begin{pmatrix} \boxed{f^*} & \boxed{-g^*} \\ \boxed{g} & \boxed{f} \end{pmatrix} \quad (\text{a})$$

By taking $\mathbf{X} = \begin{pmatrix} \boxed{f^*} & \boxed{-g^*} \\ -\boxed{F^*} & \boxed{G^*} \end{pmatrix}$, we have:

$$\begin{pmatrix} I_n & 0_n \\ \boxed{q_{01}^*} & \boxed{q_{00}} \end{pmatrix} \cdot \begin{pmatrix} \boxed{f^*} & \boxed{-g^*} \\ -\boxed{F^*} & \boxed{G^*} \end{pmatrix} = \begin{pmatrix} \boxed{f^*} & \boxed{-g^*} \\ \boxed{q_{01}^* f^*} + \boxed{q_{00}}(-\boxed{F^*}) & \boxed{q_{01}^* (-g^*)} + \boxed{q_{00}} \boxed{G^*} \end{pmatrix}$$

By definition of q_{01} and q_{00} , we have:

$$\begin{aligned} q_{01}^* f^* - q_{00} F^* &= (F^* f + G^* g) f^* - (f^2 + g^2) F^* \\ &= F^* f^2 + G^* g f^* - F^* f^2 - g^2 F^* \\ &= G^* g f^* - g^2 F^* \\ &= g(fG - gF)^* \\ &= g, \end{aligned}$$

and

$$\begin{aligned} q_{01}^* (-g^*) + q_{00} G^* &= (F^* f + G^* g)(-g^*) + (f^2 + g^2) G^* \\ &= -F^* f g^* - G^* g g^* + f^2 G^* + g^2 G^* \\ &= -F^* f g^* + f^2 G^* \\ &= f(fG - gF)^* \\ &= f. \end{aligned}$$

Besides, $\mathbf{X}$ is invertible since its determinant is equal to 1. Hence, both basis generate the same lattice.

Finally, for the last case, we show that

$$\begin{pmatrix} I_n & 0_n \\ \boxed{q_{01}} & \boxed{q_{11}} \end{pmatrix} \cdot \begin{pmatrix} \boxed{F^*} & \boxed{-G^*} \\ -\boxed{f^*} & \boxed{g^*} \end{pmatrix} = \begin{pmatrix} \boxed{F^*} & \boxed{-G^*} \\ \boxed{-G} & \boxed{F} \end{pmatrix}. \quad (\text{b})$$

Indeed,

$$\begin{aligned} q_{01} F^* - q_{11} f^* &= (F f^* + G g^*) F^* - (F^2 + G^2) f^* \\ &= F^2 f^* + G g^* F^* - f^* F^2 - G^2 f^* \\ &= G g^* F^* - G^2 f^* \\ &= G(gF - fG)^* \\ &= -G, \end{aligned}$$

and

$$\begin{aligned} q_{01} (-G^*) + q_{11} g^* &= (F f^* + G g^*) (-G^*) + (F^2 + G^2) g^* \\ &= -F f^* G^* - G g^* G^* + F^2 g^* + G^2 g^* \\ &= -F f^* G^* + F^2 g^* \\ &= F(fG - gF)^* \\ &= F. \end{aligned}$$

B Details for Theorem 1

Table 2 is obtained by multiplying both basis of Λ_{fg} in Lemma 1 by the block diagonal matrix $\mathbf{M} := \begin{pmatrix} \mathbf{Y} & 0_n \\ 0_n & \mathbf{Y} \end{pmatrix}$. Hence, both basis generate the same lattice.

Let us prove the first point. We have :

$$\begin{aligned} \det \left(\begin{pmatrix} \mathbf{Y} & 0_n \\ \mathbf{Y} \begin{smallmatrix} \overrightarrow{q_{01}} \end{smallmatrix} & \mathbf{Y} \begin{smallmatrix} \overrightarrow{q_{00}} \end{smallmatrix} \end{pmatrix} \right) &= \det \left(\mathbf{Y}^2 \begin{smallmatrix} \overrightarrow{q_{00}} \end{smallmatrix} \right) \\ &= \det \left(\begin{smallmatrix} \overrightarrow{q_{00}^{-1}} \end{smallmatrix} \begin{smallmatrix} \overrightarrow{q_{00}} \end{smallmatrix} \right) = 1. \end{aligned}$$

Finally, to prove that the good basis is orthonormal, we have to show that:

$$\begin{pmatrix} \mathbf{Y} \begin{smallmatrix} \overrightarrow{f^*} \end{smallmatrix} & \mathbf{Y} \begin{smallmatrix} \overrightarrow{-g^*} \end{smallmatrix} \\ \mathbf{Y} \begin{smallmatrix} \overrightarrow{g} \end{smallmatrix} & \mathbf{Y} \begin{smallmatrix} \overrightarrow{f} \end{smallmatrix} \end{pmatrix}^T \cdot \begin{pmatrix} \mathbf{Y} \begin{smallmatrix} \overrightarrow{f^*} \end{smallmatrix} & \mathbf{Y} \begin{smallmatrix} \overrightarrow{-g^*} \end{smallmatrix} \\ \mathbf{Y} \begin{smallmatrix} \overrightarrow{g} \end{smallmatrix} & \mathbf{Y} \begin{smallmatrix} \overrightarrow{f} \end{smallmatrix} \end{pmatrix} = \begin{pmatrix} I_n & 0_n \\ 0_n & I_n \end{pmatrix}$$

Let us compute the left hand side of the previous equation (recall that $\mathbf{Y}$ is a symmetric matrix):

$$\begin{aligned} &\begin{pmatrix} \begin{smallmatrix} \overrightarrow{f} \end{smallmatrix} & \begin{smallmatrix} \overrightarrow{g^*} \end{smallmatrix} \\ \begin{smallmatrix} \overrightarrow{-g} \end{smallmatrix} & \begin{smallmatrix} \overrightarrow{f^*} \end{smallmatrix} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{Y}^T \mathbf{Y} & 0_n \\ 0_n & \mathbf{Y}^T \mathbf{Y} \end{pmatrix} \cdot \begin{pmatrix} \begin{smallmatrix} \overrightarrow{f^*} \end{smallmatrix} & \begin{smallmatrix} \overrightarrow{-g^*} \end{smallmatrix} \\ \begin{smallmatrix} \overrightarrow{g} \end{smallmatrix} & \begin{smallmatrix} \overrightarrow{f} \end{smallmatrix} \end{pmatrix} = \\ &\begin{pmatrix} \begin{smallmatrix} \overrightarrow{f} \end{smallmatrix} & \begin{smallmatrix} \overrightarrow{g^*} \end{smallmatrix} \\ \begin{smallmatrix} \overrightarrow{-g} \end{smallmatrix} & \begin{smallmatrix} \overrightarrow{f^*} \end{smallmatrix} \end{pmatrix} \cdot \begin{pmatrix} \begin{smallmatrix} \overrightarrow{q_{00}^{-1}} \end{smallmatrix} & 0_n \\ 0_n & \begin{smallmatrix} \overrightarrow{q_{00}^{-1}} \end{smallmatrix} \end{pmatrix} \cdot \begin{pmatrix} \begin{smallmatrix} \overrightarrow{f^*} \end{smallmatrix} & \begin{smallmatrix} \overrightarrow{-g^*} \end{smallmatrix} \\ \begin{smallmatrix} \overrightarrow{g} \end{smallmatrix} & \begin{smallmatrix} \overrightarrow{f} \end{smallmatrix} \end{pmatrix} = \\ &\begin{pmatrix} \begin{smallmatrix} \overrightarrow{f} \end{smallmatrix} \begin{smallmatrix} \overrightarrow{q_{00}^{-1}} \end{smallmatrix} \begin{smallmatrix} \overrightarrow{f^*} \end{smallmatrix} + \begin{smallmatrix} \overrightarrow{g^*} \end{smallmatrix} \begin{smallmatrix} \overrightarrow{q_{00}^{-1}} \end{smallmatrix} \begin{smallmatrix} \overrightarrow{g} \end{smallmatrix} & \begin{smallmatrix} \overrightarrow{f} \end{smallmatrix} \begin{smallmatrix} \overrightarrow{q_{00}^{-1}} \end{smallmatrix} \begin{smallmatrix} \overrightarrow{-g^*} \end{smallmatrix} + \begin{smallmatrix} \overrightarrow{g^*} \end{smallmatrix} \begin{smallmatrix} \overrightarrow{q_{00}^{-1}} \end{smallmatrix} \begin{smallmatrix} \overrightarrow{f} \end{smallmatrix} \\ \begin{smallmatrix} \overrightarrow{-g} \end{smallmatrix} \begin{smallmatrix} \overrightarrow{q_{00}^{-1}} \end{smallmatrix} \begin{smallmatrix} \overrightarrow{f^*} \end{smallmatrix} + \begin{smallmatrix} \overrightarrow{f^*} \end{smallmatrix} \begin{smallmatrix} \overrightarrow{q_{00}^{-1}} \end{smallmatrix} \begin{smallmatrix} \overrightarrow{g} \end{smallmatrix} & \begin{smallmatrix} \overrightarrow{-g} \end{smallmatrix} \begin{smallmatrix} \overrightarrow{q_{00}^{-1}} \end{smallmatrix} \begin{smallmatrix} \overrightarrow{-g^*} \end{smallmatrix} + \begin{smallmatrix} \overrightarrow{f^*} \end{smallmatrix} \begin{smallmatrix} \overrightarrow{q_{00}^{-1}} \end{smallmatrix} \begin{smallmatrix} \overrightarrow{f} \end{smallmatrix} \end{pmatrix} = \\ &\begin{pmatrix} \begin{smallmatrix} \overrightarrow{f q_{00}^{-1} f^* + g^* q_{00}^{-1} g} \end{smallmatrix} & \begin{smallmatrix} \overrightarrow{f q_{00}^{-1} g^* - g^* q_{00}^{-1} f} \end{smallmatrix} \\ \begin{smallmatrix} \overrightarrow{-g q_{00}^{-1} f^* + f^* q_{00}^{-1} g} \end{smallmatrix} & \begin{smallmatrix} \overrightarrow{-g q_{00}^{-1} (-g^*) + f^* q_{00}^{-1} f} \end{smallmatrix} \end{pmatrix} = \begin{pmatrix} I_n & 0_n \\ 0_n & I_n \end{pmatrix}. \end{aligned}$$

C Details for Theorem 3

C.1 Item 1

Since $\mathbf{Y}$ is a symmetric matrix, we have $\mathbf{Y}^{-T} = \mathbf{Y}^{-1}$. Hence, we have:

$$\begin{aligned} \left\langle \begin{pmatrix} \mathbf{Y}^{-1} \begin{smallmatrix} \overrightarrow{w_1^*} \end{smallmatrix} \\ \mathbf{Y}^{-1} \begin{smallmatrix} \overrightarrow{w_2} \end{smallmatrix} \end{pmatrix}, \begin{pmatrix} \mathbf{Y} \begin{smallmatrix} \overrightarrow{f^*} \end{smallmatrix} \\ \mathbf{Y} \begin{smallmatrix} \overrightarrow{g} \end{smallmatrix} \end{pmatrix} \right\rangle &= \begin{pmatrix} \begin{smallmatrix} \overrightarrow{w_1^*} \end{smallmatrix}^T \mathbf{Y}^{-\mathbf{T}} & \begin{smallmatrix} \overrightarrow{w_2} \end{smallmatrix}^T \mathbf{Y}^{-\mathbf{T}} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{Y} \begin{smallmatrix} \overrightarrow{f^*} \end{smallmatrix} \\ \mathbf{Y} \begin{smallmatrix} \overrightarrow{g} \end{smallmatrix} \end{pmatrix} \\ &= \begin{pmatrix} \begin{smallmatrix} \overrightarrow{w_1^*} \end{smallmatrix} \\ \begin{smallmatrix} \overrightarrow{w_2} \end{smallmatrix} \end{pmatrix}^T \cdot \begin{pmatrix} \begin{smallmatrix} \overrightarrow{f^*} \end{smallmatrix} \\ \begin{smallmatrix} \overrightarrow{g} \end{smallmatrix} \end{pmatrix} \\ &= \left\langle \begin{pmatrix} \begin{smallmatrix} \overrightarrow{w_1} \end{smallmatrix} \\ \begin{smallmatrix} \overrightarrow{w_2} \end{smallmatrix} \end{pmatrix}, \begin{pmatrix} \begin{smallmatrix} \overrightarrow{f} \end{smallmatrix} \\ \begin{smallmatrix} \overrightarrow{g} \end{smallmatrix} \end{pmatrix} \right\rangle \end{aligned}$$

C.2 Item 2

We use Equation (a) defined in the proof of Lemma 1. More precisely, we have:

$$\begin{aligned} \begin{pmatrix} I_n & 0_n \\ \boxed{q_{01}^*} & \boxed{q_{00}} \end{pmatrix} \cdot \begin{pmatrix} \vec{f^*} \\ -\vec{F^*} \end{pmatrix} &= \begin{pmatrix} \vec{f^*} \\ \vec{g} \end{pmatrix} \\ \begin{pmatrix} \vec{f^*} \\ -\vec{F^*} \end{pmatrix} &= \begin{pmatrix} I_n & 0_n \\ -\boxed{q_{01}^* \cdot q_{00}^{-1}} & \boxed{q_{00}^{-1}} \end{pmatrix} \cdot \begin{pmatrix} \vec{f^*} \\ \vec{g} \end{pmatrix} \end{aligned}$$

$$\begin{pmatrix} \vec{f^*} \\ \vec{F^*} \end{pmatrix} = \begin{pmatrix} I_n & 0_n \\ \boxed{q_{01}^* \cdot q_{00}^{-1}} & -\boxed{q_{00}^{-1}} \end{pmatrix} \cdot \begin{pmatrix} \vec{f^*} \\ \vec{g} \end{pmatrix}. \quad (\text{c})$$

Therefore, we have:

$$\begin{aligned} \left\langle \mathbf{w}, \begin{pmatrix} \vec{f} \\ \vec{F} \end{pmatrix} \right\rangle &= \begin{pmatrix} \vec{w_1^*} \\ \vec{w_2^*} \end{pmatrix}^T \cdot \begin{pmatrix} \vec{f^*} \\ \vec{F^*} \end{pmatrix} \\ &= \begin{pmatrix} \vec{w_1^*} \\ \vec{w_2^*} \end{pmatrix}^T \cdot \begin{pmatrix} I_n & 0_n \\ \boxed{q_{01}^* q_{00}^{-1}} & -\boxed{q_{00}^{-1}} \end{pmatrix} \begin{pmatrix} \vec{f^*} \\ \vec{g} \end{pmatrix} \\ &= \left(\begin{pmatrix} I_n & 0_n \\ \boxed{q_{01}^* q_{00}^{-1}} & -\boxed{q_{00}^{-1}} \end{pmatrix}^T \cdot \begin{pmatrix} \vec{w_1^*} \\ \vec{w_2^*} \end{pmatrix} \right)^T \cdot \begin{pmatrix} \vec{f^*} \\ \vec{g} \end{pmatrix} \\ &= \left(\begin{pmatrix} I_n & \boxed{q_{01} q_{00}^{-1}} \\ 0_n & -\boxed{q_{00}^{-1}} \end{pmatrix} \cdot \begin{pmatrix} \vec{w_1^*} \\ \vec{w_2^*} \end{pmatrix} \right)^T \cdot \begin{pmatrix} \vec{f^*} \\ \vec{g} \end{pmatrix} \\ &= \left(\begin{pmatrix} \mathbf{Y}^{-1} & \mathbf{Y}^{-1} \boxed{q_{01} q_{00}^{-1}} \\ 0_n & -\mathbf{Y}^{-1} \boxed{q_{00}^{-1}} \end{pmatrix} \cdot \begin{pmatrix} \vec{w_1^*} \\ \vec{w_2^*} \end{pmatrix} \right)^T \cdot \begin{pmatrix} \mathbf{Y} \vec{f^*} \\ \mathbf{Y} \vec{g} \end{pmatrix} \end{aligned}$$

C.3 Item 3

We use Equation (a) defined in the proof of Lemma 1. More precisely, we have:

$$\begin{aligned} \begin{pmatrix} I_n & 0_n \\ \boxed{q_{01}^*} & \boxed{q_{00}} \end{pmatrix} \cdot \begin{pmatrix} \vec{-g^*} \\ \vec{G^*} \end{pmatrix} &= \begin{pmatrix} \vec{-g^*} \\ \vec{f} \end{pmatrix} \\ \begin{pmatrix} \vec{-g^*} \\ \vec{G^*} \end{pmatrix} &= \begin{pmatrix} I_n & 0_n \\ -\boxed{q_{01}^* \cdot q_{00}^{-1}} & \boxed{q_{00}^{-1}} \end{pmatrix} \cdot \begin{pmatrix} \vec{-g^*} \\ \vec{f} \end{pmatrix} \\ \begin{pmatrix} \vec{g^*} \\ \vec{G^*} \end{pmatrix} &= \begin{pmatrix} I_n & 0_n \\ \boxed{q_{01}^* \cdot q_{00}^{-1}} & \boxed{q_{00}^{-1}} \end{pmatrix} \cdot \begin{pmatrix} \vec{g^*} \\ \vec{f} \end{pmatrix} \\ \begin{pmatrix} \vec{g^*} \\ \vec{G^*} \end{pmatrix} &= \begin{pmatrix} 0_n & I_n \\ \boxed{q_{00}^{-1}} & \boxed{q_{01}^* \cdot q_{00}^{-1}} \end{pmatrix} \cdot \begin{pmatrix} \vec{f} \\ \vec{g^*} \end{pmatrix} \end{aligned}$$

Therefore, we have:

$$\begin{aligned}
\left\langle \mathbf{w}, \begin{pmatrix} \vec{g} \\ \vec{G} \end{pmatrix} \right\rangle &= \begin{pmatrix} \vec{w}_1^* \\ \vec{w}_2^* \end{pmatrix}^T \cdot \begin{pmatrix} \vec{g}^* \\ \vec{G}^* \end{pmatrix} \\
&= \begin{pmatrix} \vec{w}_1^* \\ \vec{w}_2^* \end{pmatrix}^T \cdot \begin{pmatrix} 0_n & I_n \\ \boxed{q_{00}^{-1}} & \boxed{q_{01}^* \cdot q_{00}^{-1}} \end{pmatrix} \cdot \begin{pmatrix} \vec{f} \\ \vec{g}^* \end{pmatrix} \\
&= \begin{pmatrix} \vec{w}_1^* \\ \vec{w}_2^* \end{pmatrix}^T \cdot \begin{pmatrix} 0_n & \mathbf{Y}^{-1} \\ \boxed{q_{00}^{-1}} \mathbf{Y}^{-1} & \boxed{q_{01}^* \cdot q_{00}^{-1}} \mathbf{Y}^{-1} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{Y} \vec{f} \\ \mathbf{Y} \vec{g}^* \end{pmatrix} \\
&= \left(\begin{pmatrix} 0_n & \mathbf{Y}^{-1} \boxed{q_{00}^{-1}} \\ \mathbf{Y}^{-1} & \mathbf{Y}^{-1} \boxed{q_{01}^* \cdot q_{00}^{-1}} \end{pmatrix} \begin{pmatrix} \vec{w}_1^* \\ \vec{w}_2^* \end{pmatrix} \right)^T \cdot \begin{pmatrix} \mathbf{Y} \vec{f} \\ \mathbf{Y} \vec{g}^* \end{pmatrix} \\
&= \left(\begin{pmatrix} 0_n & \mathbf{Y}^{-1} \boxed{q_{00}^{-1}} \\ \mathbf{Y}^{-1} & \mathbf{Y}^{-1} \boxed{q_{01}^* \cdot q_{00}^{-1}} \end{pmatrix} \begin{pmatrix} \vec{w}_1^* \\ \vec{w}_2^* \end{pmatrix} \right)^T \cdot \begin{pmatrix} \mathbf{Y} \vec{f} \\ \mathbf{Y} \vec{g}^* \end{pmatrix}
\end{aligned}$$

C.4 Item 4

Equation (b) defined in the proof of Lemma 1 give:

$$\begin{aligned}
\begin{pmatrix} I_n & 0_n \\ \boxed{q_{01}} & \boxed{q_{11}} \end{pmatrix} \cdot \begin{pmatrix} \vec{F}^* \\ -\vec{f}^* \end{pmatrix} &= \begin{pmatrix} \vec{F}^* \\ -\vec{G} \end{pmatrix} \\
\begin{pmatrix} I_n & 0_n \\ \boxed{-q_{01}} & \boxed{q_{11}} \end{pmatrix} \cdot \begin{pmatrix} \vec{F}^* \\ \vec{f}^* \end{pmatrix} &= \begin{pmatrix} \vec{F}^* \\ \vec{G} \end{pmatrix}
\end{aligned}$$

Using Equation (c) from the proof of item 2 of Theorem 3, we have:

$$\begin{aligned}
\begin{pmatrix} I_n & 0_n \\ \boxed{-q_{01}} & \boxed{q_{11}} \end{pmatrix} \cdot \begin{pmatrix} \boxed{q_{01}^* \cdot q_{00}^{-1}} & \boxed{-q_{00}^{-1}} \\ I_n & 0_n \end{pmatrix} \cdot \begin{pmatrix} \vec{f}^* \\ \vec{g} \end{pmatrix} &= \begin{pmatrix} \vec{F}^* \\ \vec{G} \end{pmatrix} \\
\begin{pmatrix} \boxed{q_{01}^* q_{00}^{-1}} & \boxed{-q_{00}^{-1}} \\ \boxed{-q_{01} q_{01}^* q_{00}^{-1} + q_{11}} & \boxed{q_{01} q_{00}^{-1}} \end{pmatrix} \cdot \begin{pmatrix} \vec{f}^* \\ \vec{g} \end{pmatrix} &= \begin{pmatrix} \vec{F}^* \\ \vec{G} \end{pmatrix}
\end{aligned}$$

Therefore, we have:

$$\begin{aligned}
\left\langle \mathbf{w}, \begin{pmatrix} \vec{F} \\ \vec{G} \end{pmatrix} \right\rangle &= \begin{pmatrix} \vec{w}_1 \\ \vec{w}_2 \end{pmatrix}^T \cdot \begin{pmatrix} \vec{F} \\ \vec{G} \end{pmatrix} \\
&= \begin{pmatrix} \vec{w}_1^* \\ \vec{w}_2^* \end{pmatrix}^T \cdot \begin{pmatrix} \vec{F}^* \\ \vec{G} \end{pmatrix} \\
&= \begin{pmatrix} \vec{w}_1^* \\ \vec{w}_2^* \end{pmatrix}^T \cdot \begin{pmatrix} \boxed{q_{01}^* q_{00}^{-1}} & \boxed{-q_{00}^{-1}} \\ \boxed{-q_{01} q_{01}^* q_{00}^{-1} + q_{11}} & \boxed{q_{01} q_{00}^{-1}} \end{pmatrix} \cdot \begin{pmatrix} \vec{f}^* \\ \vec{g} \end{pmatrix} \\
&= \begin{pmatrix} \vec{w}_1^* \\ \vec{w}_2^* \end{pmatrix}^T \cdot \begin{pmatrix} \boxed{q_{01}^* q_{00}^{-1}} \mathbf{Y}^{-1} & \boxed{-q_{00}^{-1}} \mathbf{Y}^{-1} \\ \boxed{-q_{01} q_{01}^* q_{00}^{-1} + q_{11}} \mathbf{Y}^{-1} & \boxed{q_{01} q_{00}^{-1}} \mathbf{Y}^{-1} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{Y} \vec{f}^* \\ \mathbf{Y} \vec{g} \end{pmatrix} \\
&= \left(\begin{pmatrix} \mathbf{Y}^{-1} \boxed{q_{00}^{-1}} \boxed{q_{01}} & \mathbf{Y}^{-1} \left(-\boxed{q_{00}^{-1}} \boxed{q_{01}} \boxed{q_{01}^*} + \boxed{q_{11}} \right) \\ \mathbf{Y}^{-1} \boxed{-q_{00}^{-1}} & \mathbf{Y}^{-1} \boxed{q_{00}^{-1}} \boxed{q_{01}^*} \end{pmatrix} \cdot \begin{pmatrix} \vec{w}_1^* \\ \vec{w}_2^* \end{pmatrix} \right)^T \cdot \begin{pmatrix} \mathbf{Y} \vec{f}^* \\ \mathbf{Y} \vec{g} \end{pmatrix}
\end{aligned}$$

C.5 Details for Corollary 3

The result follows from the following equalities:

$$\begin{aligned}
&\begin{pmatrix} \vec{w}_1 \\ \vec{w}_2 \end{pmatrix}^T \cdot \begin{pmatrix} \mathbf{Y}^{-1} & \mathbf{Y}^{-1} \boxed{q_{01} q_{00}^{-1}} \\ 0_n & -\mathbf{Y}^{-1} \boxed{q_{00}^{-1}} \end{pmatrix}^T \cdot \begin{pmatrix} \mathbf{Y}^{-1} & \mathbf{Y}^{-1} \boxed{q_{01} q_{00}^{-1}} \\ 0_n & -\mathbf{Y}^{-1} \boxed{q_{00}^{-1}} \end{pmatrix} \cdot \begin{pmatrix} \vec{w}_1^* \\ \vec{w}_2^* \end{pmatrix} \\
&= \begin{pmatrix} \vec{w}_1 \\ \vec{w}_2 \end{pmatrix}^T \cdot \begin{pmatrix} \mathbf{Y}^{-1} & 0_n \\ \mathbf{Y}^{-1} \boxed{q_{01}^* q_{00}^{-1}} & -\mathbf{Y}^{-1} \boxed{q_{00}^{-1}} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{Y}^{-1} & \mathbf{Y}^{-1} \boxed{q_{01} q_{00}^{-1}} \\ 0_n & -\mathbf{Y}^{-1} \boxed{q_{00}^{-1}} \end{pmatrix} \cdot \begin{pmatrix} \vec{w}_1^* \\ \vec{w}_2^* \end{pmatrix} \\
&= \begin{pmatrix} \vec{w}_1 \\ \vec{w}_2 \end{pmatrix}^T \cdot \begin{pmatrix} \boxed{q_{00}} & \boxed{q_{01}} \\ \boxed{q_{01}^*} & \boxed{q_{11}} \end{pmatrix} \cdot \begin{pmatrix} \vec{w}_1^* \\ \vec{w}_2^* \end{pmatrix} \\
&= \|\mathbf{w}\|_Q^2
\end{aligned}$$

$$\begin{aligned}
&\begin{pmatrix} \vec{w}_1^* \\ \vec{w}_2^* \end{pmatrix}^T \cdot \begin{pmatrix} 0_n & \mathbf{Y}^{-1} \boxed{q_{00}^{-1}} \\ \mathbf{Y}^{-1} & \mathbf{Y}^{-1} \boxed{q_{01}^* \cdot q_{00}^{-1}} \end{pmatrix}^T \cdot \begin{pmatrix} 0_n & \mathbf{Y}^{-1} \boxed{q_{00}^{-1}} \\ \mathbf{Y}^{-1} & \mathbf{Y}^{-1} \boxed{q_{01}^* \cdot q_{00}^{-1}} \end{pmatrix} \cdot \begin{pmatrix} \vec{w}_1 \\ \vec{w}_2 \end{pmatrix} \\
&= \begin{pmatrix} \vec{w}_1^* \\ \vec{w}_2^* \end{pmatrix}^T \cdot \begin{pmatrix} 0_n & \mathbf{Y}^{-1} \\ \mathbf{Y}^{-1} \boxed{q_{00}^{-1}} & \mathbf{Y}^{-1} \boxed{q_{01} \cdot q_{00}^{-1}} \end{pmatrix} \cdot \begin{pmatrix} 0_n & \mathbf{Y}^{-1} \boxed{q_{00}^{-1}} \\ \mathbf{Y}^{-1} & \mathbf{Y}^{-1} \boxed{q_{01}^* \cdot q_{00}^{-1}} \end{pmatrix} \cdot \begin{pmatrix} \vec{w}_1 \\ \vec{w}_2 \end{pmatrix} \\
&= \begin{pmatrix} \vec{w}_1^* \\ \vec{w}_2^* \end{pmatrix}^T \cdot \begin{pmatrix} \boxed{q_{00}} & \boxed{q_{01}^*} \\ \boxed{q_{01}} & \boxed{q_{11}} \end{pmatrix} \cdot \begin{pmatrix} \vec{w}_1 \\ \vec{w}_2 \end{pmatrix} \\
&= \|\mathbf{w}\|_Q^2
\end{aligned}$$

References

- [ACD⁺18] Martin R. Albrecht, Benjamin R. Curtis, Amit Deo, Alex Davidson, Rachel Player, Eamonn W. Postlethwaite, Fernando Virdia, and Thomas Wunderer. Estimate all the LWE, NTRU schemes! In Dario Catalano and Roberto De Prisco, editors, *SCN 18*, volume 11035 of *LNCS*, pages 351–367. Springer, Cham, September 2018.
- [ADPS16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In Thorsten Holz and Stefan Savage, editors, *USENIX Security 2016*, pages 327–343. USENIX Association, August 2016.
- [AGVW17] Martin R. Albrecht, Florian Göpfert, Fernando Virdia, and Thomas Wunderer. Revisiting the expected cost of solving uSVP and applications to LWE. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 297–322. Springer, Cham, December 2017.
- [Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th ACM STOC*, pages 99–108. ACM Press, May 1996.
- [APv25] Bill Allombert, Alice Pellet-Mary, and Wessel P. J. van Woerden. Cryptanalysis of rank-2 module-LIP: A single real embedding is all it takes. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part II*, volume 15602 of *LNCS*, pages 184–212. Springer, Cham, May 2025.
- [AR24] Martin R. Albrecht and Joe Rowell. Scaling lattice sieves across multiple machines. *CiC*, 1(4):11, 2024.
- [BBD⁺23a] Joppe W. Bos, Olivier Bronchain, Léo Ducas, Serge Fehr, Yu-Hsuan Huang, Thomas Pornin, Eamonn W. Postlethwaite, Thomas Prest, Ludo N. Pulles, and Wessel van Woerden. HAWK. Technical report, National Institute of Standards and Technology, 2023. available at <https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures>.
- [BBD⁺23b] Joppe W. Bos, Olivier Bronchain, Léo Ducas, Serge Fehr, Yu-Hsuan Huang, Thomas Pornin, Eamonn W. Postlethwaite, Thomas Prest, Ludo N. Pulles, and Wessel van Woerden. HAWK specification version 1.0.1. *NIST submission to the call for post-quantum signatures*, 2023.
- [BSW16] Shi Bai, Damien Stehlé, and Weiqiang Wen. Improved reduction from the bounded distance decoding problem to the unique shortest vector problem in lattices. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *ICALP 2016*, volume 55 of *LIPIcs*, pages 76:1–76:12. Schloss Dagstuhl, July 2016.
- [CDH⁺20] Cong Chen, Oussama Danba, Jeffrey Hoffstein, Andreas Hülsing, Joost Rijneveld, John M. Schanck, Peter Schwabe, William Whyte, Zhenfei Zhang, Tsunekazu Saito, Takashi Yamakawa, and Keita Xagawa. NTRU. Technical report, National Institute of Standards and Technology, 2020. available at <https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions>.

- [CJC25] Jinzheng Cao, Haodong Jiang, and Qingfeng Cheng. Refined attack on LWE with hints: Constructing lattice via gaussian elimination. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part I*, volume 16000 of *LNCS*, pages 385–416. Springer, Cham, August 2025.
- [CME⁺25] Clémence Chevnard, Guilhem Mureau, Thomas Espitau, Alice Pellet-Mary, Heorhii Pliatsok, and Alexandre Wallet. A reduction from hawk to the principal ideal problem in a quaternion algebra. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part II*, volume 15602 of *LNCS*, pages 154–183. Springer, Cham, May 2025.
- [CN11] Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better lattice security estimates. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 1–20. Springer, Berlin, Heidelberg, December 2011.
- [DDGR20] Dana Dachman-Soled, Léo Ducas, Huijing Gong, and Mélissa Rossi. LWE with side information: Attacks and concrete security estimation. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 329–358. Springer, Cham, August 2020.
- [DGHK23] Dana Dachman-Soled, Huijing Gong, Tom Hanson, and Hunter Kippen. Revisiting security estimation for LWE with hints from a geometric perspective. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 748–781. Springer, Cham, August 2023.
- [DLP14] Léo Ducas, Vadim Lyubashevsky, and Thomas Prest. Efficient identity-based encryption over NTRU lattices. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 22–41. Springer, Berlin, Heidelberg, December 2014.
- [DPPvW22] Léo Ducas, Eamonn W. Postlethwaite, Ludo N. Pulles, and Wessel P. J. van Woerden. Hawk: Module LIP makes lattice signatures fast, compact and simple. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part IV*, volume 13794 of *LNCS*, pages 65–94. Springer, Cham, December 2022.
- [DSDG⁺22] Dana Dachman-Soled, Léo Ducas, Huijing Gong, Mélissa Rossi, and Thibault Feneuil. leaky-lwe-estimator, 2022. Accessed at commit 0a9caf8 on 2022-11-14.
- [Duc18] Léo Ducas. Shortest vector from lattice sieving: A few dimensions for free. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 125–145. Springer, Cham, April / May 2018.
- [DvW22] Léo Ducas and Wessel P. J. van Woerden. On the lattice isomorphism problem, quadratic forms, remarkable lattices, and cryptography. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part III*, volume 13277 of *LNCS*, pages 643–673. Springer, Cham, May / June 2022.
- [GMRR22] Morgane Guereau, Ange Martinelli, Thomas Ricosset, and Mélissa Rossi. The hidden parallelepiped is back again: Power analysis attacks on falcon. *IACR TCHES*, 2022(3):141–164, 2022.
- [GR24] Morgane Guereau and Mélissa Rossi. A not so discrete sampler: Power analysis attacks on HAWK signature scheme. *IACR TCHES*, 2024(4):156–178, 2024.

- [HR14] Ishay Haviv and Oded Regev. On the lattice isomorphism problem. In Chandra Chekuri, editor, *25th SODA*, pages 391–404. ACM-SIAM, January 2014.
- [KH18] Suhri Kim and Seokhie Hong. Single Trace Analysis on Constant Time CDT Sampler and Its Countermeasure. *Applied Sciences*, 8(10), 2018.
- [LDK⁺22] Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrede Lepoint, Peter Schwabe, Gregor Seiler, Damien Stehlé, and Shi Bai. CRYSTALS-DILITHIUM. Technical report, National Institute of Standards and Technology, 2022. available at <https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022>.
- [LJL82] A. K. Lenstra, H. W. Lenstra Jr., and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.
- [LJPW24] Hengyi Luo, Kaijie Jiang, Yanbin Pan, and Anyu Wang. Cryptanalysis of rank-2 module-LIP with symplectic automorphisms. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part IV*, volume 15487 of *LNCS*, pages 359–385. Springer, Singapore, December 2024.
- [LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, Berlin, Heidelberg, May / June 2010.
- [MLB15] Artur Mariano, Thijs Laarhoven, and Christian H. Bischof. Parallel (probable) lock-free HashSieve: A practical sieving algorithm for the SVP. In *Proceedings of the 44th International Conference on Parallel Processing (ICPP 2015)*, pages 590–599. IEEE Computer Society, 2015.
- [MN23] Alexander May and Julian Nowakowski. Too many hints - when LLL breaks LWE. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part IV*, volume 14441 of *LNCS*, pages 106–137. Springer, Singapore, December 2023.
- [MPM26] Guilhem Mureau and Alice Pellet-Mary. Dimension reduction for SVP in Hawk: A trace-zero approach. Cryptology ePrint Archive, Paper 2026/1560, 2026.
- [MPPW24] Guilhem Mureau, Alice Pellet-Mary, Georgii Pliatsok, and Alexandre Wallet. Cryptanalysis of rank-2 module-LIP in totally real number fields. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part VII*, volume 14657 of *LNCS*, pages 226–255. Springer, Cham, May 2024.
- [MUTS22] Soundes Marzougui, Vincent Ulitzsch, Mehdi Tibouchi, and Jean-Pierre Seifert. Profiling side-channel attacks on Dilithium: A small bit-fiddling leak breaks it all. Cryptology ePrint Archive, Report 2022/106, 2022.
- [PDPvW23] Ludo Pulles, Léo Ducas, Eamonn W. Postlethwaite, and Wessel van Woerden. hawk-aux, 2023. Accessed at commit 02b1f0a on 2026-04-10.
- [PFH⁺22] Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. FALCON. Technical report, National Institute of Standards and Technology, 2022. available at <https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022>.

- [SAB⁺22] Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, Damien Stehlé, and Jintai Ding. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2022. available at <https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022>.
- [SE94] Claus-Peter Schnorr and Martin Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical Programming*, 66(1-3):181–199, 1994.
- [Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th FOCS*, pages 124–134. IEEE Computer Society Press, November 1994.
- [SW26] Zygimantas Straznickas and Stephen A. Weis. HAWK- n key recovery reduces to SVP in dimension $n/2 + 1$. *Cryptology ePrint Archive*, Paper 2026/1593, 2026.
- [ZDY25] Ziyu Zhao, Jintai Ding, and Bo-Yin Yang. Sieving with streaming memory access. *IACR TCHES*, 2025(2):362–384, 2025.
- [ZLYW23] Shiduo Zhang, Xiuhan Lin, Yang Yu, and Weijia Wang. Improved power analysis attacks on falcon. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part IV*, volume 14007 of *LNCS*, pages 565–595. Springer, Cham, April 2023.