Gulf Journal of Mathematics
Vol 19, Issue 2 (2025) 384-397

ISSN: 2309-4966
https://doi.org/10.56947/gjom.v19i2.2778

AN ALGEBRAIC KEY-EXPOSURE ATTACK ON THE DIGITAL SIGNATURE HAWK

NOUR-EDDINE RAHMANI[1*], TAOUFIK SERRAJ[2] and MOULAY CHRIF ISMAILI [3]

Abstract. Hawk is a post-quantum signature scheme. The security and efficiency of its most recent version are primarily based on the structured variant of a problem known as lattice isomorphism problem (the module variant Module-LIP). We propose an attack on Hawk when one of the four coefficients of the signing key is leaked. This attack combines algebraic methods with side-channel exploitations. The proposed attack running time is a polynomial in the degree of the cyclotomic number field.

1. Introduction

After Peter Shor's paper in 1994 [35], the need for alternatives to public cryptographic protocols based on classical problems, such as integer factorization and discrete logarithms, became evident. The past two decades have been marked by intense efforts from worldwide researchers, leading to many fascinating new designs.

The National Institute of Standards and Technologies (NIST) requested fresh proposals in 2016 to develop standards that can be implemented on ordinary computers and can withstand quantum attacks. NIST selected four algorithms to standardize in 2023: CRYSTALS-KYBER *CRYSTALS-KYBER* [9] (ML-KEM [26]) , a method for secure keys; and three digital signatures: CRYSTALS-DILITHIUM [15] (ML-DSA [27]), FALCON [18] (FN-DSA, future FIPS 206), and SPHINCS [8] (SLH-DSA [28]).

In [13], Ducas and van Woerden proposed the application of the lattice isomorphism problem in cryptography. The module variant of the problem is difficult, which the security of of HAWK is based on, meaning the hardness of the lattice isomorphism problem (LIP) over modules [14] which is a special case of the (unstructured) LIP. This allows efficient computations and reducing communications and storage costs as in the other lattice-based protocols.

Practically, its interaction with the attack as well as with the cryptographic primitive depends not just on its mathematical formulation but also on its implementation and the physical attributes of the device used for the primitive's

Date: Received: Mar 18, 2025; Accepted: Apr 6, 2025.

* Corresponding author.

2020 *Mathematics Subject Classification.* Primary 46L55; Secondary 44B20.

Key words and phrases. Cryptanalysis, digital signature, HAWK, lattices, lattice isomorphism problem, side channel attacks.

realization. Side-channel attacks arise in this setting as a result of the unintended leakage of sensitive data by cryptographic hardware when following through with the protocol. The present paper presents the approach to exploiting the leakage of a single algebraic integer coefficient of a signing key, i.e., a coefficient belonging to the ring of integers of a power of two cyclotomic field.

The rest of the paper is organized as follows: Section 2 recalls some notions on cyclotomic fields, lattices, the module lattice isomorphism problem, the inner product, and norms. Section 3 briefly recalls HAWK algorithms. In Section 4, we prove a linear dependence result that will be necessary to lead the attack proposed in the next section. Our proposed attack is presented and discussed in Section 5. Finally, a conclusion is provided.

2. Preliminaries and Notations

2.1. **Number Fields.** A number field L is an extension of $\mathbb{Q}$ and is denoted as $\mathbb{Q}[X]/(P(X))$ for an irreducible polynomial $P \in \mathbb{Q}[X]$. It has the structure of a $\mathbb{Q}$-vector space of dimension n, where n is the degree of P. The ring of integers of a number field is the set of all elements α of L such that there exists a monic irreducible polynomial $T \in \mathbb{Z}[X]$ that annihilates α, i.e., $T(\alpha) = 0$, and it is denoted by $\mathcal{O}_L$.

Each number field comes with a set of automorphisms, known as its Galois group, which is defined as

$$G := \mathrm{Gal}(L/\mathbb{Q}) = \{\sigma \in \mathrm{Aut}(L) \mid \sigma(x) = x, \forall x \in \mathbb{Q}\}.$$

From algebraic number theory, the group G is finite and has n automorphisms, where n is the degree of the number field L.

The algebraic norm of an element a of L is computed as

$$\mathrm{Tr}_L(a) = \sum_{\sigma \in G} \sigma(a),$$

and its trace is

$$\prod_{\sigma \in G} \sigma(a).$$

The discriminant of the number field is given by

$$\Delta_L = \det\left(\mathrm{Tr}\left(a_i a_j\right)_{1 \le i,j \le n}\right),$$

where $\{a_1, \ldots, a_n\}$ is a basis of L as a $\mathbb{Q}$-vector space.

2.2. **Cyclotomic number fields.** The HAWK signature scheme [14, 13] integrates number fields and their integer rings. For $n = 2^m$, where $m \in \{8, 9, 10\}$, the scheme utilises the cyclotomic number field $L = \mathbb{Q}[X]/(X^n + 1)$ and its corresponding ring of integers $\mathcal{O}_L = \mathbb{Z}[X]/(X^n + 1)$. One of the reasons of using these class of number fields is due to existence of fast multiplication algorithms to multiply elements of $\mathcal{O}_L$, one of the most used algorithms is the Number Theoretic Transform (NTT). The late algorithm is used in most of lattice-based constructions such as Kyber, Dilithium and NTRU and also HAWK, the scheme we are attacking in our paper (see for example the NTT algorithm review [33] for more

information on the techniques). Another reason is to ensure that multiplying two elements of $\mathcal{O}_L$, does not grow so fast, this is required because of euclidean norm conditions in verifications procedure in lattice-based digital signatures and correction of the error when it required for lattice based encryption schemes.

2.3. **Lattices and Lattices Problems.** In this subsection, we provide the essential math facts required for the paper. We also discuss lattices and the challenging problems associated with them.

Definition 2.1 (Lattice)**.** A lattice $\mathcal{L}$ consists of all vectors in $\mathbb{R}^n$ obtained by all linear integer combinations of the set $B = \{b_i | 1 \leq i \leq r\}$ where b_i's are linearly indepandant vectors in $\mathbb{R}^n$, and we have :

$$\mathcal{L} := \mathcal{L}(B) = \left\{ \sum_{i=1}^{r} z_i b_i, z_i \in \mathbb{Z} \right\}$$

The natural numbers r are referred to as rank of lattice, and n as dimension, while B represents the basis. It is called a full rank lattice when $r = n$.

One of the important invariants that comes with lattices is its volume; it measures how dense the lattice is in the ambient vector space $\mathbb{R}^n$.

Definition 2.2. The volume of the lattice is defined by :

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

for any basis B of the lattice $\mathcal{L}$.

The matrix B, with columns $\mathbf{b}_i$, is a basis of $\mathcal{L}$. All bases of $\mathcal{L}$ can be changed by multiplying on the right by an matrix U of determinant ± 1, which belongs to the general linear group $\mathrm{GL}_n(\mathbb{Z})$. The shortest vector problem is given a lattice, to find a non-zero vector which has minimal Euclidean norm.

Definition 2.3 (The shortest vector problem (SVP))**.** Given a basis B of a lattice $\mathcal{L}$, find a non-zero vector $v \in \mathcal{L}$ such that $\|v\| = \lambda_1(\mathcal{L}) := \min_{vin\mathcal{L}\setminus\{0\}} \|v\|$. v is called a *shortest vector* of the lattice $\mathcal{L}$.

Another important lattice problem is the closest vector problem where we are given a basis of a lattice and a target vector which is non-lattice vector in the real spanned vector space of the lattice, to find a nearest lattice vector to the that target.

Definition 2.4 (The closest vector problem (CVP))**.** Given a basis B of a lattice $\mathcal{L}$ and a target vector $t \in \mathrm{span}(\mathcal{L})$ find a lattice vector $u \in \mathcal{L}$ such that $\|u - t\| = \mathrm{dist}(\mathcal{L}, t)$.

In general, given a good basis of the lattice, the CVP could be considered easy. There are many algorithms in the literature to solve the closest vector problem, and most famous ones are Babai nearest plane [6] algorithm and Kannan's embedding [21] technique. Both of the algorithms requires having a good basis to output a correct answer, here where lattice reduction plays an important role [23, 4, 34] . The complexity of computing a good basis from a given basis of a

lattice it depends about the situation, for example if the input basis is already reduced then no need for lattice reduction. This is the case in our study, the lattice we are solving the closest vector problem already has a good basis, and hence solving the problem is easier in our case and enabling our attack to be realised.

Now, we proceed to state the Lattice Isomorphism Problem as described in [13, 25].

Definition 2.5 (The Lattice Isomorphism Problem [13, 25])**.** Two lattices $\mathcal{L}$ and $\mathcal{L}'$, with respective bases B and B', are considered isomorphic if there exists an orthogonal transformation O such that $\mathcal{L}' = O\mathcal{L}$. Equivalently, in matrix form, $\mathcal{L}(B)$ and $\mathcal{L}(B')$ are isomorphic if and only if there exists a unimodular matrix $U \in \mathrm{GL}_n(\mathbb{Z})$ satisfying:

$$B' = OBU.$$

The computational problem associated with this definition involves finding either such an orthogonal matrix O or a unimodular matrix U.

Rewriting the Lattice Isomorphism Problem in terms of quadratic forms for simplicity. The definite positive quadratic form $G = B^\top B \in \mathrm{S}_n^>(\mathbb{R})$ is the Gram matrix associated with a basis $B \in \mathrm{GL}_n(\mathbb{R})$. Two quadratic forms $Q, Q' \in \mathrm{S}_n^>(\mathbb{R})$ are said to be congruent if there exists $U \in \mathrm{GL}_n(\mathbb{Z})$ such that $Q' = U^\top QU$.

Definition 2.6 (wc-sLIP$_\mathrm{Q}$ [25](Definition 2.2))**.** Let $Q \in \mathrm{S}_n^>(\mathbb{R})$. The phworst-case search Lattice Isomorphism Problem with parameter Q, denoted wc-sLIP$_Q$, is:

- **Input:** A quadratic form $Q' \in \mathrm{S}_n^>(\mathbb{R})$ that is congruent to Q.
- **Goal:** Find a unimodular matrix $U \in \mathrm{GL}_n(\mathbb{Z})$ such that

$$Q' = U^\top Q\, U.$$

By using the Cholesky decomposition, one can show that wc-sLIP$_B$ and wc-sLIP$_Q$ are polynomial-time equivalent [13, 25].

2.4. **Module Lattices.** Module lattices derive their security strength from the reductions between worst-case and average-case scenarios. In particular, solving a Module-LIP problem for a given public key Q inherently implies resolving lattice isomorphism for any module lattice contained in $\mathcal{O}_L^2$. This problem is widely believed to resist quantum attacks [13, 14].

Module lattices, a structured generalisation of traditional lattices, are defined over specific algebraic fields.

Using the ring $R_n := \mathcal{O}_L = \mathbb{Z}[X]/(X^n + 1)$ from the cyclotomic number field $\mathbb{Q}(\zeta_{2n})$, a rank-2 module lattice $M \subset \mathcal{O}_L^2$ can be expressed through a basis B as:

$$M = \{Bx \mid x \in \mathcal{O}_L^2\}. \tag{2.1}$$

2.5. **Inner Product and Norms.** Within these CM-fields, complex conjugation is given by:

$$f^\star = f_0 - f_{n-1}X - \cdots - f_1X^{n-1}.$$

An inner product on L is introduced as:

$$\langle f, g\rangle = \frac{1}{n}\mathrm{Tr}(f^\star g),$$

where $\mathrm{Tr}(\cdot)$ denotes the trace map. The Euclidean norm associated with this inner product are defined as:

$$\|f\| = \sqrt{\langle f, f\rangle}.$$

For a matrix $B \in L^{r\times r}$, the Q-inner product and its corresponding norm are expressed as:

$$\langle f, g\rangle_Q = \frac{1}{n}\mathrm{Tr}(f^\star Q g), \quad \|f\|_Q = \sqrt{\langle f, f\rangle_Q},$$

where Q is defined as $b^\star B$ where $b^\star$ is the transpose conjugate of B.

The absolute value of the algebraic norm of f is the determinant of its negacyclic associated matrix :

$$|\mathcal{N}_L(f)| = \prod_{\sigma\in G} |\sigma(f)| = |\det(\mathrm{rot}(f))|$$

The matrix associated defines a lattice; this type of lattice is called an ideal lattice. Since we are working in the power-of-two cyclotomic lattice we consider the volume of the lattice associated to an algebraic integer[1] equals the absolute value of the determinant of its rotational matrix.

2.6. **The Centered Binomial Distribution.** The centered binomial distribution plays a crucial role in the design of hawk, we recall its definition and properties.

Definition 2.7 (Centered Binomial Distribution)**.** Each coordinate d_i of the error vector follows a centered binomial distribution:

$$C_\eta = \sum_{i=1}^{\eta}(x_i - y_i), \quad x_i, y_i \sim \text{Bernoulli}(1/2).$$

The variance of C_η is:

$$\mathrm{Var}(C_\eta) = \eta/2.$$

Thus, the standard deviation of each coordinate is:

$$\sigma = \sqrt{\eta/2}.$$

3. HAWK Algorithms

In this section, we present the algorithms that form the core of the HAWK signature scheme, including key generation, signing, and verification processes.

[1]This correspondence is under the umbrella of coefficient embedding lattice, there is another embedding in the literature called the Minkowski embedding. For our purpose we are satisfied by the coefficient embedding

3.1. **HAWK Key Generation.** The private key consists of a basis $B = \begin{pmatrix} f & F \\ g & G \end{pmatrix}$ that satisfies the NTRU-like relation:

$$fG - gF = 1,^{2} \tag{3.1}$$

where $f, g, F, G \in R$. The public key is the Gram matrix:

$$Q = B^{\star}B. \tag{3.2}$$

The basis is chosen to ensure a balance between security and computational efficiency. Coefficients of f and g are sampled from a centered binomial distribution with standard deviation η to achieve compactness and security [14].

3.2. **HAWK Signature Generation.** The signing process transforms a message into a secure signature using the following steps:

(1) Hash the message m to a vector $h \in R^2$ using a cryptographic hash function H:
$$h = H(m\|\text{salt}) \in R^2. \tag{3.3}$$
(2) Compute the target vector $t = Bh \mod 2$.
(3) Sample x from the discrete Gaussian distribution $D_{2\mathbb{Z}^{2n}+t,\sigma}$.
(4) the signature is computed is follows :
$$s = \frac{1}{2}(h - B^{-1}x). \tag{3.4}$$

The signature s is compressed for storage efficiency, reducing s modulo a small bound determined by σ_{verify}, ensuring compactness without loss of verification accuracy [14].
Remark on x. Here x is a secret vector, and what we are proving in this paper is knowing of the two coefficients of this vector is sufficient to recover the signing key.

3.3. **HAWK Verification.** The verification process ensures the authenticity of a signature using the following steps:

(1) Recompute h and $w = h - 2s$.
(2) Verify that:
$$\|w\|_Q^2 \leq \sigma_{\text{verify}}^2 \cdot 2n. \tag{3.5}$$

This ensures that the signature is valid and correctly associated with the public key [14, 13].

3.4. **HAWK's Security.**
Worst-Case to Average-Case Reductions. HAWK leverages reductions from worst-case instances of module-LIP to average-case instances, ensuring robust protection against adversaries, we refer the reader to [13, 14] for more details.

[2]Remember this equation is $\pmod{X^n + 1}$

4. Linear Dependence

In this section, we establish the relationship between B and Q over $\mathcal{O}_L$, the ring of integers of the CM-number field L.

Proposition 4.1. *Let* $B = \begin{pmatrix} f & F \\ g & G \end{pmatrix}$ *be a matrix in* $\mathrm{GL}_2(\mathcal{O}_L)$, $d_B = \det(B)$, *and let*

$$Q = \begin{pmatrix} a & b^\star \\ b & c \end{pmatrix} = B^\star B,$$

then

$$F = \frac{fb^\star - g^\star d_B}{a}, \quad G = \frac{f^\star d_B + gb^\star}{a}.$$

Proof. The proof is analogous for F and G, so we provide the details for F only.

Substituting into the numerator on the right-hand side of F by $b^\star = f^\star F + g^\star G$ and $d_B = fG - gF$, we get:

$$fb^\star - g^\star d_B = ff^\star F + fg^\star G - g^\star fG + g^\star gF = (ff^\star + gg^\star)F = aF.$$

This concludes the proof. □

An important case for HAWK is when $d_C = 1$ we have the following corollary:

Corollary 4.2. *Let* $B = \begin{pmatrix} f & F \\ g & G \end{pmatrix}$ *be a matrix in* $\mathrm{GL}_2(\mathcal{O}_L)$, $\det(B) = 1$, *and let*

$$Q = \begin{pmatrix} a & b^\star \\ b & c \end{pmatrix} = B^\star B,$$

then

$$F = \frac{fb^\star - g^\star}{a}, \quad G = \frac{f^\star + gb^\star}{a}.$$

5. One coefficient of the signing-key exposure attack

Now using the previous proposition and corollary, we can derive an attack when side information appears from side channels. In the this section, we prove without loss of generality that even when only g is leaked, a full recovery is still possible in polynomial time, using the LLL algorithm. The corollary shows that a divides $fb^\star - g^\star$ and a divides $f^\star + gb^\star$, that is there is $F, G \in \mathcal{O}_L$ such that $Fa = fb^\star - g^\star$ and $Ga = f^\star + gb^\star$. Since a, b and c are public, given f or g, to retrieve g or f respectively, one must solve either one of the diophantine equations:

$$Fa = fb^\star - g^\star \text{ or } Ga = f^\star + gb^\star,$$

for unknowns F, g or G, f respectively. The two diophantine equations are the same $Xa = Y + \alpha b^\star$ for unknowns X and Y, whereas the known coefficient α is f or g depend on the situation when the attack knows f or g respectively. Without loss of generality, this can be extended in the same manner for any given coefficient of B. Since this is just a redundancy, we describe the attack only in the case of giving, g and we have to retrieve f, and clearly by the corollary 4.2, we compute F and G, therefore we are able to construct the whole signing key.

5.1. **Toward solving the diophantine equation $va = f^\star + gb^\star$ via lattices.** To retrieve f given g, one must solve one of the previous equations:

$$Fa = fb^\star - g^\star \text{ or } Ga = f^\star + gb^\star.$$

In our assumed scenario, to get f from Q and g we will try to solve $Ga = f^\star + gb^\star$ using lattices. This choice is because that this equation it can be seen as a closest vector problem instance with target $gb^\star$ and the lattice constructed by a and the error term is $f^\star$.

The following key lemma will help us to prove that in our case we are able to solve the closest vector problem in this setting.

Lemma 5.1. *For two algebraic integers f and g of $\mathcal{O}_L$, with each coefficient is sampled from the centred binomial distribution of parameter η and let $a = ff^\star + gg^\star$. We have:*

(1) *The expected Euclidean norm of f and g is:*

$$\mathbb{E}\left[\|f\|\right] = \mathbb{E}\left[\|g\|\right] \approx O(\sqrt{n\eta})$$

(2) *The expected Euclidean norm of a is:*

$$\mathbb{E}\left[\|a\|\right] \approx O(n\eta)$$

(3) *The expected algebraic norm of a is:*

$$\mathbb{E}\left[\mathcal{N}(a)\right] \approx O((n\eta)^n)$$

Proof. Each coefficient of $f = \sum_{i=0}^{n-1} f[i]X^i$ or g is sampled identically and independently from the centered binomial distribution, and since $\|f\|^2 = \sum_{i=0}^{n-1} f[i]^2$, We have:

(1) $\mathbb{E}\left[\|f\|^2\right] = \sum_{i=0}^{n-1} \mathbb{E}[f_i^2] = \sum_{i=0}^{n-1} \frac{\eta}{2} = \frac{n\eta}{2}$, therefore $\mathbb{E}\left[\|f\|^2\right] = O(n\eta)$. Using the fact that $\mathbb{E}[\mathcal{X}^2] \geq \mathbb{E}[\mathcal{X}]^2$ and taking square root we prove the first claim, i.e. $\mathbb{E}\left[\|f\|\right] = O(\sqrt{n\eta})$.

(2) The same procedure was applied after considering the convolutions and estimating for each coefficient, a_k of a we get $\mathbb{E}[a_k^2] \approx 2n\eta^2$, making that $\mathbb{E}[\|a\|^2] \approx 2n^2\eta^2$ and hence $\mathbb{E}\left[\|a\|\right] = O(n\eta)$.

(3) This follows from the previous point and the formula of the algebraic norm, and since we are working in a power-of-two cyclotomic field, we have $|\mathcal{N}_L(a)| = \prod_{\sigma \in G} |\sigma(a)| \approx \|a\|^n \approx O((n\eta)^n)$ from the previous point.

□

We give an empirical comparison with theoretical bounds given in the previous lemma. The diagram in figure 1 summarizes what the lemma predicts:

The Gaussian heuristic estimates the length of a shortest non-zero vector in a lattice of dimension n as follows:

$$\lambda_1(\mathcal{L}) \approx \sqrt{\frac{n}{2\pi e}} \mathrm{vol}(\mathcal{L})^{\frac{1}{n}}$$

For the principal ideal lattice $\mathcal{L}(a)$ and the previous lemma, we have the following heuristic:

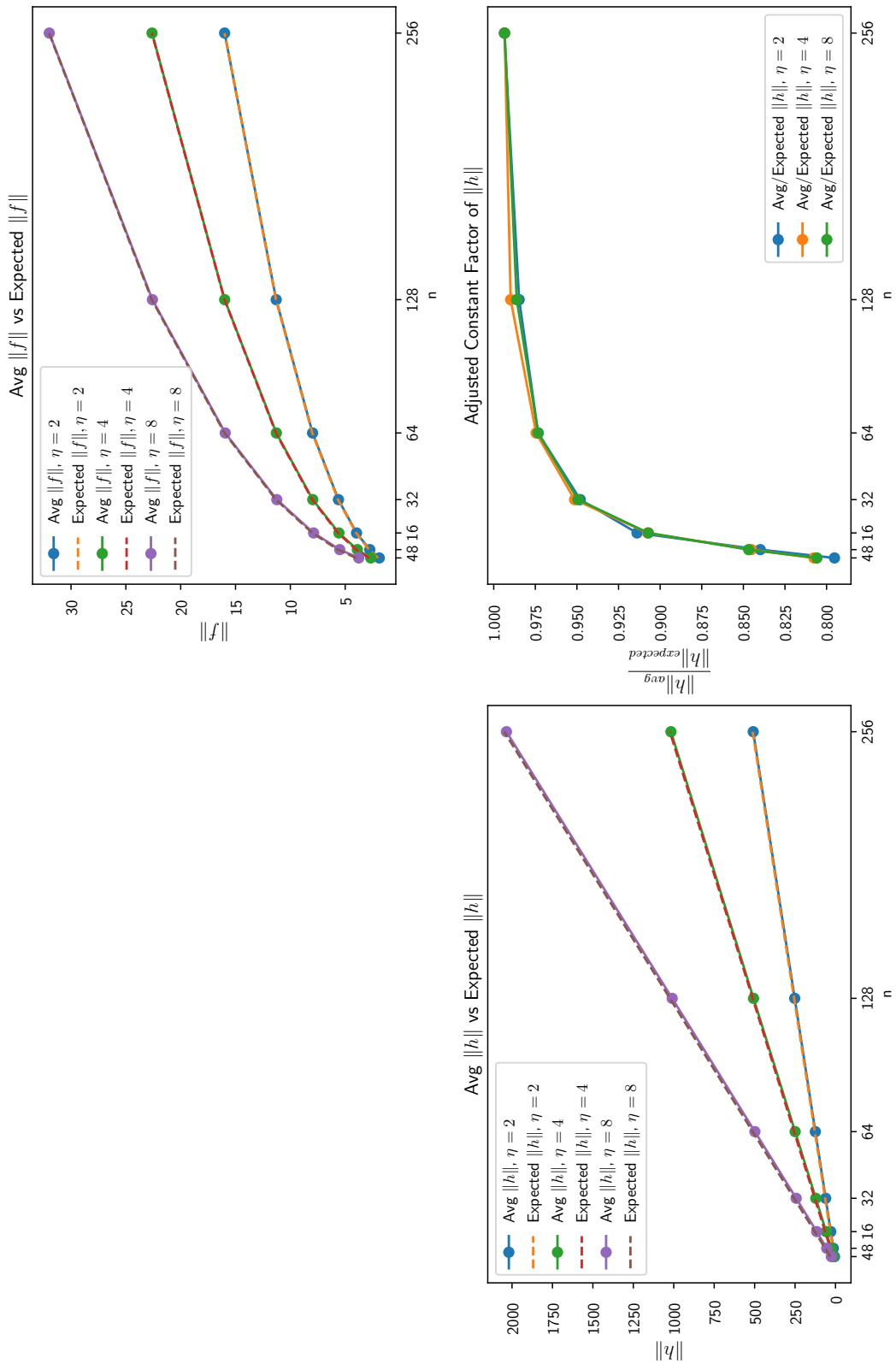

FIGURE 1. Comparison of Theoretical and Empirical Norms

$$\lambda_1(\mathcal{L}(a)) \approx \sqrt{\frac{n}{2\pi e}}\,|\mathcal{N}_L(a)|^{\frac{1}{n}} \approx \sqrt{\frac{n}{2\pi e}}O(n\eta) \approx O(n^{\frac{3}{2}}\eta)$$

From the previous discussion, on the one hand, we heuristically predict that $\mathrm{vec}(a) = \mathrm{vec}(a^\star)$ is a short vector in the ideal lattice generated by a, since it is much smaller than what the Gaussian heuristic predicts (and we are given the vector of a), on the other hand, the error term $\mathrm{vec}(f)$ is smaller than $\mathrm{vec}(a)$ and grows slower than $\mathrm{vec}(a)$ do as n or η grow.

The equation $g^\star b = aG^\star + f$ cane be seen in the language of lattices as:

$$\mathrm{vec}(g^\star b) = \mathrm{rot}(a) \cdot \mathrm{vec}(G^\star) + \mathrm{vec}(f).$$

Since a is public, we can decode the vector $\mathrm{vec}(g^\star b)$ to a lattice vector u of the lattice $\mathcal{L}(a)$, in polynomial time using Kannan's embedding or Babai's nearest plane algorithm, and since a is of small euclidean norm, the output should be f with overwhelming probability. Another important remark, under the same setting, is that a shortest vector estimated length and the error estimated length share n and η, this implies that for any η under the same construction and assumptions, the attack is valid. We have the following algorithm:

Algorithm 1 Recovering f from (a, b, g) using LLL and Kannan's embedding

Input: a, b: Public elements of the public key of HAWK
Input: g: a leaked algebraic coefficient of the secret key from the HAWK construction
Input: η: Parameter controlling the noise distribution
Output: f

Compute the vector representation of $g^\star b$:
  $\mathrm{vec}_g = \mathrm{vec}(g^\star b)$
Construct the lattice basis using Kannan's embedding:
  $M = \begin{bmatrix} \mathrm{Rot}(a) & \mathrm{vec}_g \\ 0 & 1 \end{bmatrix}$
Apply the LLL algorithm to M to obtain a reduced basis:
  Reduced_Basis = LLL(M)
Extract the first vector from the reduced basis:
  short_vector = Reduced_Basis[0]
Verify and return f:
if short_vector is of the form $(\mathrm{vec}(f), \pm 1)^T$ **then**
    return $\mp\mathrm{vec}(f)$
else
    Report failure (should not happen with overwhelming probability)
end if

Hence, we have the following theorem, ensuring that algorithm 1 terminates and return the correct answer.

Theorem 5.2. *For any η, and for any n degree of a power of two cyclotomic number field, where f, g are sampled as in the construction of HAWK from a*

centred binomial distribution of parameter η, there is a polynomial-time algorithm 1 that outputs f given a, b, and g.

Proof. From the lemma 5.1, on average the lattice $\mathcal{L}(a)$ is already reduced, and consider the following lattice by constructed by the Kannan's embedding technique:

$$M = \begin{bmatrix} \mathrm{Rot}(a) & \mathrm{vec}(g^\star b) \\ 0 & 1 \end{bmatrix}.$$

The lattice defined by M contains the vector $(\mathrm{vec}(f), -1)^T$ that is predicted to have a small Euclidean norm, since $M \cdot (\mathrm{vec}(G^\star), -1) = (\mathrm{vec}(aG^\star) - \mathrm{vec}(g^\star b), -1)^T = (\mathrm{vec}(f), -1)^T$. Running the LLL algorithm, we get a shortest lattice vector $(\mathrm{vec}(f), -1)^T$ or $(-\mathrm{vec}(f), 1)^T$. Since the LLL runs in polynomial time, then the algorithm runs in polynomial time. □

Now retrieving a second coefficient is possible given one of the four coefficients, and by corollary 4.2, the whole signing key can be reconstructed. This does not take a long time in practice.

5.2. **Performance of the attack using Kannan's Embedding.** This subsection presents the results of the tests for the previously mentioned attack. The test was conducted on all different parameters in the documentation of HAWK, in particular for $n = 1024$, we find a second coefficient of the coefficients of B given one another of its coefficients in about 90 seconds on average of 5 instances using the official demonstrative repository of HAWK implemented in SageMath:

https://github.com/ludopulles/hawk-aux/blob/main/code/hawk.sage

Our implementation which loads the previous scripts can be found In

https://github.com/ndrahmani/hawk_key_exposure_attack

The following table 1 summarizes the behaviour for different degrees provided in the official documentation of HAWK, using the LLL function implemented in SageMath 10.5 [32].

Degree of Cyclotomic Field	256	512	1024
Running Time (seconds)	0.7918	6.4686	81.9059

TABLE 1. Running time of the attack for different degrees of power-of-two cyclotomic fields.

From the previous table we see that our attack is efficient when a leakage is provided, hence we gave an efficient method to exploit extra information.

We also have run the experiment on larger η (For $\eta = 3200$ this correspond as input `sigma_kgs` $= 40$) for cyclotomic field of degree $n = 256$ with 100 tests and all the tests the other coefficient is successfully retrieved. This validates our theoretical analysis in practice. The average running time over 100 tests was 0.84 seconds, indicating a slight increase in execution time. This demonstrates that the effect of η is smaller than that of the number field degree as it grows. Consequently, increasing η cannot serve as a mathematical countermeasure to our

proposed attack that exploits side information. Therefore, securing the scheme against physical leakage should be a priority.

6. Conclusion

In this paper we gave a new approach to benefit from an exposure to attack HAWK using leakage from the key generation or during the signing process. The attack runs in polynomial time in the degree of the cyclotomic number field. We let to extend the attack when less information about signing key, or signing secret coefficients is available as an open problem.

References

1. Alkim, E., Ducas, L., Pöppelmann, T., & Schwabe, P. (2016). Post-quantum key exchange - a new hope. SEC'16: Proceedings of the 25th USENIX Conference on Security Symposium, 327–343. https://doi.org/10.5555/3241094.3241120
2. Abdelmoumen, K., Ben-Azza, H., & Otmani, A. (2016). Pade approximants and key lattices for decoding alternant codes. Gulf Journal of Mathematics, 4(4). https://doi.org/10.56947/gjom.v4i4.278
3. Albrecht, M. R., Gheorghiu, V., Postlethwaite, E. W., & Schanck, J. M. (2020). Estimating quantum speedups for lattice sieves. In Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part II (pp. 583-613). Springer International Publishing. https://doi.org/10.1007/978-3-030-64834-3_20
4. Albrecht, M. R., Ducas, L., Herold, G., Kirshanova, E., Postlethwaite, E. W., & Stevens, M. (2019). The general sieve kernel and new records in lattice reduction. In Y. Ishai & V. Rijmen (Eds.), Advances in Cryptology – EUROCRYPT 2019 (Vol. 11477, pp. 717–746). Springer. https://doi.org/10.1007/978-3-030-17656-3_25
5. Azizi, A., Jerrari, I., Zekhnini, A., & Talbi, M. (2016). Units of some abelian imaginary number fields of type (2, 4). Gulf Journal of Mathematics, 4(4). https://doi.org/10.56947/gjom.v4i4.275
6. Babai, L. (1986). On Lovász' lattice reduction and the nearest lattice point problem. Combinatorica, 6, 1–13. https://doi.org/10.1007/BF02579403
7. Bennett, H., Ganju, A., Peetathawatchai, P., & Stephens-Davidowitz, N. (2023). Just how hard are rotations of $\mathbb{Z}^n$? Algorithms and cryptography with the simplest lattice. In C. Hazay & M. Stam (Eds.), Advances in Cryptology – EUROCRYPT 2023 (Vol. 14008, pp. 643–673). Springer. https://doi.org/10.1007/978-3-031-30589-4_9
8. Bernstein, D. J., et al. (2015). SPHINCS: Practical stateless hash-based signatures. In E. Oswald & M. Fischlin (Eds.), Advances in Cryptology – EUROCRYPT 2015. EUROCRYPT 2015. Lecture Notes in Computer Science, vol. 9056 (pp. 368–397). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-46800-5_15
9. Bos, J., et al. (2018). CRYSTALS - Kyber: A CCA-secure module-lattice-based KEM. In 2018 IEEE European Symposium on Security and Privacy (EuroS&P) (pp. 353–367). https://doi.org/10.1109/EuroSP.2018.00032
10. Cohen, G., Shpilka, A., & Tal, A. (2017). On the degree of univariate polynomials over the integers. Combinatorica, 37(4), 419–464. https://doi.org/10.1007/s00493-015-2987-0
11. Das, D., Hoffstein, J., Pipher, J., Whyte, W., & Zhang, Z. (2020). Modular lattice signatures, revisited. Designs, Codes and Cryptography, 88(3), 505–532. https://doi.org/10.1007/s10623-019-00694-x
12. Dachman-Soled, D., Ducas, L., Gong, H., & Rossi, M. (2020). LWE with side information: Attacks and concrete security estimation. In D. Micciancio & T. Ristenpart

(Eds.), Advances in Cryptology – CRYPTO 2020. CRYPTO 2020. Lecture Notes in Computer Science, vol. 12171 (pp. 329–358). Springer, Cham. https://doi.org/10.1007/978-3-030-56880-1_12
13. Ducas, L., & van Woerden, W. (2022). On the lattice isomorphism problem, quadratic forms, remarkable lattices, and cryptography. In O. Dunkelman & S. Dziembowski (Eds.), Advances in Cryptology – EUROCRYPT 2022 (pp. 643–673). Springer, Cham. https://doi.org/10.1007/978-3-031-07082-2_23
14. Ducas, L., Postlethwaite, E. W., Pulles, L. N., & van Woerden, W. (2022). Hawk: Module LIP makes lattice signatures fast, compact and simple. In S. Agrawal & D. Lin (Eds.), Advances in Cryptology – ASIACRYPT 2022. ASIACRYPT 2022. Lecture Notes in Computer Science, vol. 13794 (pp. 65–94). Springer, Cham. https://doi.org/10.1007/978-3-031-22972-5_3
15. Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., & Stehlé, D. (2018). CRYSTALS-Dilithium: A lattice-based digital signature scheme. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2018(1), 238–268. https://doi.org/10.13154/tches.v2018.i1.238-268
16. Eum, S., Lee, M., & Seo, H. (2024). Optimizing HAWK signature scheme performance on ARMv8. Applied Sciences, 14(19), 8647. https://doi.org/10.3390/app14198647
17. Fenzi, G., Moghaddas, H., & Nguyen, N.K. (2024). Lattice-based polynomial commitments: Towards asymptotic and concrete efficiency. Journal of Cryptology, 37(3). https://doi.org/10.1007/s00145-024-09511-8
18. Fouque, P. A., et al. (2019). Falcon: Fast-Fourier lattice-based compact signatures over NTRU. Proceeding NIST PQC Standardization Conference. https://api.semanticscholar.org/CorpusID:231637439
19. Gentry, C., Peikert, C., & Vaikuntanathan, V. (2008). Trapdoors for hard lattices and new cryptographic constructions. Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing, 197–206. https://doi.org/10.1145/1374376.1374407
20. Guerreau, M., & Rossi, M. (2024). A not so discrete sampler: Power analysis attacks on HAWK signature scheme. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2024(4), 156–178. https://doi.org/10.46586/tches.v2024.i4.156-178
21. Kannan, R. (1983). Improved algorithms for integer programming and related lattice problems. In Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 193–206. https://doi.org/10.1145/800061.808749
22. Langlois, A., & Stehlé, D. (2015). Worst-case to average-case reductions for module lattices. Design, Codes and Cryptography, 75, 565–599. https://doi.org/10.1007/s10623-014-9938-4
23. Lenstra, A. K., Lenstra, H. W., & Lovász, L. (1982). Factoring polynomials with rational coefficients. Mathematische Annalen, 261(4), 515–534. https://doi.org/10.1007/BF01457454
24. Luo, H., Jiang, K., Pan, Y., & Wang, A. (2024). Cryptanalysis of Rank-2 Module-LIP with Symplectic Automorphisms. In K. M. Chung & Y. Sasaki (Eds.), Advances in Cryptology – ASIACRYPT 2024. ASIACRYPT 2024. Lecture Notes in Computer Science, vol. 15487 (pp. 359–385). Springer, Singapore. https://doi.org/10.1007/978-981-96-0894-2_12
25. Mureau, G., Pellet-Mary, A., Pliatsok, G., & Wallet, A. (2024). Cryptanalysis of Rank-2 Module-LIP in Totally Real Number Fields. In M. Joye & G. Leander (Eds.), Advances in Cryptology – EUROCRYPT 2024. EUROCRYPT 2024. Lecture Notes in Computer Science, vol. 14657 (pp. 226–255). Springer, Cham. https://doi.org/10.1007/978-3-031-58754-2_9
26. National Institute of Standards and Technology. (2024). Module-Lattice-Based Key Encapsulation Mechanism Standard. Federal Information Processing Standards Publication (FIPS) NIST FIPS 203. Washington, D.C. https://doi.org/10.6028/NIST.FIPS.203

27. National Institute of Standards and Technology. (2024). Module-Lattice-Based Digital Signature Standard. Federal Information Processing Standards Publication (FIPS) NIST FIPS 204. Washington, D.C. https://doi.org/10.6028/NIST.FIPS.204
28. National Institute of Standards and Technology. (2024). Stateless Hash-Based Digital Signature Standard. Federal Information Processing Standards Publication (FIPS) NIST FIPS 205. Washington, D.C. https://doi.org/10.6028/NIST.FIPS.205
29. Hoffstein, J., Pipher, J., & Silverman, J. H. (1998). NTRU: A ring-based public key cryptosystem. In J. P. Buhler (Ed.), Algorithmic Number Theory (pp. 267–288). Springer, Berlin, Heidelberg. https://doi.org/10.1007/BFb0054868
30. Peikert, C. (2016). A decade of lattice cryptography. Foundations and Trends in Theoretical Computer Science, 10(4), 283–424. https://doi.org/10.1561/0400000074
31. Regev, O. (2025). An efficient quantum factoring algorithm. Journal of ACM, 72(1). https://doi.org/10.1145/3708471
32. The Sage Developers. (2024). SageMath, the Sage mathematics software system (Version 10.5). https://www.sagemath.org
33. Satriawan, A., Syafalni, I., Mareta, R., Anshori, I., Shalannanda, W., & Barra, A. (2023). Conceptual review on number theoretic transform and comprehensive review on its implementations. IEEE Access, 11, 70288-70316. https://doi.org/10.1109/ACCESS.2023.3294446
34. Schnorr, C. P. (1987). A hierarchy of polynomial time lattice basis reduction algorithms. Theoretical Computer Science, 53(2), 201–224. https://doi.org/10.1016/0304-3975(87)90064-8
35. Shor, P. W. (1994). Algorithms for quantum computation: Discrete logarithms and factoring. Proceedings 35th Annual Symposium on Foundations of Computer Science, 124–134. https://doi.org/10.1109/SFCS.1994.365700
36. Washington, L. C. (1997). Introduction to cyclotomic fields. Graduate Texts in Mathematics, Springer, New York. https://doi.org/10.1007/978-1-4612-1934-7
37. Zhao, Z., & Ding, J. (2023). Practical improvements on BKZ algorithm. In Cyber Security, Cryptology, and Machine Learning (pp. 273–284). https://doi.org/10.1007/978-3-031-34671-2_19

[1*] ACSA Laboratory, Department of Mathematics, Faculty of Sciences, Mohammed I University, Oujda, Morocco
Email address: nour-eddine.rahmani@ump.ac.ma

[2] ACSA Laboratory, Department of Mathematics, Faculty of Sciences, Mohammed I University, Oujda, Morocco
Email address: t.serraj@ump.ac.ma

[3] ACSA Laboratory, Department of Mathematics, Faculty of Sciences, Mohammed I University, Oujda, Morocco
Email address: mc.ismaili@ump.ac.ma