HAWK with Hint: Algebraic Key Recovery from Side-Channel Leakage

Byoungchan Chi*
bchan052415@gmail.com
Korea University
Seoul, Republic of Korea

Changmin Lee†
changminlee@korea.ac.kr
Korea University
Seoul, Republic of Korea

Inhun Lee*
dldlsgns99@korea.ac.kr
Korea University
Seoul, Republic of Korea

Abstract

We investigate how partial physical leakage can be amplified through public algebraic relations, using HAWK as a concrete case study of implementation security in an algebraically structured post-quantum signature scheme. HAWK is a lightweight, floating-point-free lattice-based signature scheme whose discrete Gaussian sampler can expose side-channel information during signing. We formalize such leakage as *HAWK with Hint* and study full-coefficient, exact-sign, and noisy-sign leakage. Our polynomial-time recovery algorithms combine sampler hints with public signing information and HAWK's Gram-matrix public-key structure, showing that complete sampler-output recovery is not necessary for full key recovery.

For HAWK-1024, we obtain the following results under the stated threat models. With full coefficient leakage, a single signature suffices for secret-key recovery by directly solving the induced linear system. On the HAWK-1024 reference implementation, we identify all 2,048 coefficient-sampler invocations in each of 14 measured signing traces, extract all 28,672 coefficient signs without error, and recover the fixed key using the Category II attack. In separate controlled experiments over 100 independently generated keys, Category II achieves 100% success using 14 exact-sign signatures in approximately 157 seconds. Under the independent sign-error model, Category III achieves 100% success using 420 signatures at a 10% error rate and 13,397 signatures at a 40% error rate. The latter setting requires approximately 178 seconds. These results show that coarse and even erroneous sampler leakage can become sufficient for full key recovery when combined with HAWK's public algebraic structure.

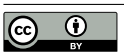
CCS Concepts

• **Security and privacy → Side-channel analysis and countermeasures**.

Keywords

Post-Quantum Cryptography, Digital Signature, HAWK, Side-Channel Analysis

*These authors contributed equally to this work.
†Corresponding author.



This work is licensed under a Creative Commons Attribution 4.0 International License.
CCS '26, The Hague, Netherlands.
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2871-6/2026/11
https://doi.org/10.1145/XXXXXX.XXXXXX

ACM Reference Format:
Byoungchan Chi, Changmin Lee, and Inhun Lee. 2026. HAWK with Hint: Algebraic Key Recovery from Side-Channel Leakage. In *Proceedings of the 2026 ACM SIGSAC Conference on Computer and Communications Security (CCS '26), November 15–19, 2026, The Hague, Netherlands.* ACM, New York, NY, USA, 22 pages. https://doi.org/10.1145/XXXXXX.XXXXXX

1 Introduction

Quantum computing has intensified the need for post-quantum cryptography (PQC), prompting the development and evaluation of new key-encapsulation and signature schemes. HAWK is a lattice-based signature scheme with a lightweight, floating-point-free design [2]. HAWK advanced to Round 3 of NIST's Additional Digital Signatures process. Recent public-key cryptanalysis substantially lowered its estimated key-recovery cost [21], and HAWK was subsequently withdrawn from the standardization process. The recent cryptanalysis is public-key-only and requires neither signing executions nor physical leakage. Our work considers a distinct setting in which physical leakage arises during signing. HAWK's lightweight design supports efficient implementations, including on constrained platforms. However, avoiding floating-point arithmetic does not by itself protect its discrete Gaussian sampler against timing, power, or electromagnetic leakage.

Implementation leakage from discrete Gaussian sampling is particularly relevant to lattice-based signatures, where ephemeral sampler outputs can interact with public signing information to reveal secret-key information. Prior work has shown that branching patterns, table accesses, and arithmetic operations can expose sampled values and, in several schemes, support full key recovery attacks [5–8, 12, 24]. In Falcon [17], for example, observations related to signs, table accesses, and rejection behavior have been translated into practical attacks [8, 11, 13, 24]. These results establish Gaussian-sampler leakage as a recurring threat across lattice signatures. However, the corresponding key-recovery methods depend on scheme-specific relations between sampler outputs, public signing information, and the secret key. They therefore do not directly transfer to HAWK.

HAWK consequently requires a distinct analysis of how sampler leakage interacts with its algebraic structure. Although HAWK and Falcon are both lattice-based hash-and-sign schemes that employ Gaussian sampling, HAWK relies on a different underlying problem and represents its public key as a Gram matrix. HAWK also applies a signature-wise symmetry-breaking step. Existing Falcon recovery equations therefore do not directly yield a HAWK secret key. Prior side-channel work on HAWK showed that sign, parity, and increment-related information can be extracted and used to reconstruct most or all of the sampler output, thereby enabling key recovery [9]. However, that work targeted near-complete or

complete sampler-output reconstruction rather than systematic key recovery from sign-only or noisy-sign leakage. It therefore remained unclear whether algebraic key recovery is feasible when only coarse or error-prone sign observations are available.

This gap motivates a broader implementation security question. We ask how much information about a sampler output is sufficient for full key recovery when the leakage is combined with public algebraic relations. In practice, an adversary may recover complete coefficient values under favorable measurement conditions, but may instead obtain only signs or error-corrupted observations. We formalize these regimes as *HAWK with Hint* and develop recovery methods for full-coefficient, exact-sign, and noisy-sign leakage. Our concrete recovery techniques exploit HAWK's Gram-matrix public key, its signing relations, and its symmetry-breaking procedure. They are therefore specific to HAWK's algebraic structure. In this sense, HAWK serves as a concrete case study of implementation security in an algebraically structured post-quantum signature scheme. The broader methodological contribution is to show that coarse or noisy physical leakage need not reveal complete sampler outputs to be security-critical. Such leakage can become sufficient for full key recovery when combined with suitable scheme-specific relations between the public transcript and the secret key. Our results demonstrate this phenomenon with exact sign leakage in the measured HAWK-1024 experiment and with noisy sign observations under the independent error model analyzed here.

1.1 Contributions

With this distinction in mind, our main contributions are summarized below.

- **Formalization of *HAWK with Hint*.** We introduce a formal framework for modeling side-channel information in HAWK through an adversary's partial knowledge of the DGS output. We consider three leakage categories: (i) full coefficient recovery, (ii) exact sign information, and (iii) noisy sign information.
- **HAWK-specific algebraic identities for integrating sampler leakage.** We derive algebraic relations linking leakage from HAWK's DGS to the secret basis. In particular, these identities exploit HAWK's Gram-matrix public-key representation to translate sign information into explicit algebraic constraints. They provide a systematic way to incorporate sampler leakage and form the common algebraic foundation of Categories I, II, and III.
- **Category-specific polynomial-time key recovery algorithms.** For each leakage category, we design a polynomial-time key recovery algorithm. In Category I, full coefficient leakage from a single signature suffices for exact algebraic recovery. In Category II, exact sign leakage leads to a structured linear program solvable by an interior-point method. In Category III, we develop a recovery algorithm that remains effective under independent bit-flip noise with rate $p < \frac{1}{2}$.
- **Implementation and experimental validation.** We implement the recovery algorithms and evaluate them using measured and controlled leakage. In Category I, full coefficient leakage from a single signature suffices for exact key recovery. On the HAWK-1024 reference implementation, we collect 14 measured traces, each covering a complete `sig_gauss()` execution invoked during signing. We identify all 2,048 coefficient-sampler invocations in each trace, extract all 28,672 coefficient signs without error, and directly supply the extracted signs to the Category II attack, which recovers the fixed key. This validates the sampler-stage trace-to-key pipeline for one fixed key. In separate controlled experiments over 100 independently generated keys, Category II achieves 100% success for HAWK-1024 using 14 exact-sign signatures in approximately 157 seconds. Under simulated independent sign errors, Category III achieves 100% success using 420 signatures at a 10% error rate and 13,397 signatures at a 40% error rate. The latter setting requires approximately 178 seconds.

1.2 Related Work

Discrete Gaussian sampling is a well-known source of side-channel leakage in lattice-based signatures. For CDT-style Gaussian samplers [16], Kim and Hong [12] showed that even a constant-time implementation can leak enough information in a single power trace to recover sampled values. Falcon [17] has been studied especially intensively in this direction. Karabulut and Aysu [11] demonstrated a key-recovery attack on an unprotected Falcon implementation using electromagnetic leakage. Guerreau et al. [8] later showed that leakage from Falcon's BaseSampler can be used to select useful signatures and recover the secret key. Zhang et al. [24] improved this line of attack with a more efficient statistical recovery method and showed that leakage of the random sign bit in Falcon's integer Gaussian sampler is sufficient for practical key recovery under their experimental setting. Similar side-channel weaknesses have also been studied for BLISS and for NTRU-lattice hash-and-sign signatures [5–7].

More recently, Guerreau and Rossi [9] analyzed HAWK's Gaussian sampler and showed that sign, parity, and increment-related leakage can be used to reconstruct most or all of the sampled vector. In the strongest case, exact sampler outputs from two signatures are sufficient to reconstruct the secret basis with high probability. Their recovery therefore relies on near-complete or complete sampler-output reconstruction rather than sign-only or noisy-sign leakage.

In parallel, several works have studied how partial leakage, often called *hints*, can help lattice attacks. Dachman-Soled et al. [3] introduced a general framework for using different kinds of hints in primal attacks on LWE. Wu et al. [23] studied how such hints can also be used in dual attacks, while May and Nowakowski [15] showed that many perfect or modular hints can make LWE key recovery possible with only LLL reduction. Lu et al. [14] extended this direction by considering independent hints on both the secret and the error terms and by making the lattice construction more efficient. Hermelink et al. [10] proposed a general framework for LWE-based schemes that can handle noisy or probabilistic leakage. Beyond LWE, Shao et al. [20] adapted the hint-based viewpoint to NTRU lattices and showed that partial coefficient leakage can reduce the cost of recovering Falcon secret keys.

Most closely related to our work are recent analyses of HAWK under strong leakage or key-exposure assumptions. Rahmani et al. [18] study an algebraic key-exposure setting in which one of the four ring elements of the secret basis is leaked and show that algebraic relations between the public matrix and the secret basis can be used to recover the secret key. Their subsequent work [19]

recovers the secret basis from an exact sampler output and its corresponding public signing information from one signature. Both that work and our Category I therefore have the same one-signature requirement under exact sampler-output leakage. Their approach first recovers one column of the secret basis and then reconstructs the remaining column from public-key relations, whereas our Category I expresses recovery of the complete basis through a unified algebraic identity. The main significance of this identity is that it also provides the foundation for Categories II and III, which cover weaker exact-sign and noisy-sign leakage models.

A comparison of the leakage assumptions, signature requirements, noise robustness, and key-recovery methods of prior HAWK attacks is provided in Section 5.5 and Table 6.

2 HAWK

In this section, we present the HAWK signature scheme and establish the algebraic notation that will be used throughout its description and analysis. Its design relies on NTRU-type bases, Hermitian forms, and trace-based inner products, which together provide the algebraic framework for both key generation and verification. We begin by introducing the necessary notation and conventions.

2.1 Notation

Let $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{C}$ denote the integers, rational numbers, and complex numbers, respectively. We work over the negacyclic ring

$$R_n \coloneqq \mathbb{Z}[X]/(X^n+1), \qquad K_n \coloneqq \mathbb{Q}[X]/(X^n+1),$$

where $n = 2^m$. Bold lowercase letters denote vectors over R_n. All indices start at 0.

Hermitian involution and trace pairing. For

$$a(X) = \sum_{i=0}^{n-1} a[i]X^i \in R_n,$$

define

$$a^\star \coloneqq a(X^{-1}) \bmod (X^n+1),$$

which corresponds to negacyclic coefficient reversal, namely

$$a^\star[0] = a[0], \qquad a^\star[i] = -a[n-i] \quad \text{for } 1 \le i < n.$$

For a matrix $M = (m_{ij}) \in R_n^{r\times r}$, where $R_n^{r\times r}$ denotes the set of $r \times r$ matrices over R_n, define

$$M^\star \coloneqq (m_{ji}^\star).$$

For $u \in K_n$, let

$$\mathrm{T}(u) \coloneqq \tfrac{1}{n}\mathrm{Tr}_{K_n/\mathbb{Q}}(u)$$

denote the normalized field trace, where $\mathrm{Tr}_{K_n/\mathbb{Q}}$ is the field trace from K_n to $\mathbb{Q}$, i.e., the sum of the algebraic conjugates of u. We use the trace pairing

$$\langle f, g\rangle \coloneqq \mathrm{T}(f^\star g), \qquad \langle \mathbf{u}, \mathbf{v}\rangle \coloneqq \mathrm{T}(u_0^\star v_0 + u_1^\star v_1),$$

for $f, g \in R_n$ and $\mathbf{u} = (u_0, u_1), \mathbf{v} = (v_0, v_1) \in R_n^2$. In particular,

$$\langle f, g\rangle = (f^\star g)[0],$$

that is, the pairing extracts the constant coefficient of $f^\star g$. For any Hermitian matrix $Q \in R_n^{2\times 2}$, we define the associated quadratic form by

$$\|\mathbf{w}\|_Q^2 \coloneqq \mathrm{T}(\mathbf{w}^\star Q\, \mathbf{w}).$$

Complex embedding. Let

$$\zeta = e^{i\pi/n}$$

be a primitive $2n$-th root of unity. The canonical embedding

$$\sigma : K_n \to \mathbb{C}^n$$

is defined by

$$\sigma(a) \coloneqq \big(a(\zeta), a(\zeta^3), \ldots, a(\zeta^{2n-1})\big) = \big(a(\zeta^{2k+1})\big)_{k=0}^{n-1}.$$

This embedding is $\mathbb{Q}$-linear, meaning that it preserves addition and multiplication by rational scalars, and extends componentwise to vectors and matrices over R_n. Moreover, the Hermitian involution corresponds to complex conjugation under this embedding:

$$\sigma(a^\star) = \overline{\sigma(a)}.$$

Hence the trace pairing can be written as

$$\langle f, g\rangle = \frac{1}{n}\sum_{k=0}^{n-1} \sigma(f^\star)_k\, \sigma(g)_k = \frac{1}{n}\sum_{k=0}^{n-1} \overline{\sigma(f)_k}\, \sigma(g)_k.$$

We also write

$$R_n^{\mathbb{R}} \coloneqq \mathbb{R}[X]/(X^n+1),$$

which is used for real-valued operations such as norms and square roots during signature generation.

Vectorization and the lift operator. Fix the power basis

$$(1, X, \ldots, X^{n-1}),$$

and let

$$\mathrm{vec}(a) \in \mathbb{Z}^n$$

denote the coefficient vector of $a \in R_n$. Let $S \in \mathbb{Z}^{n\times n}$ be the negacyclic shift matrix

$$S = \begin{pmatrix} 0 & 0 & \cdots & 0 & -1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix},$$

which represents multiplication by X in R_n, i.e.,

$$\mathrm{vec}(Xa) = S\,\mathrm{vec}(a).$$

For

$$a = \sum_{i=0}^{n-1} a[i]X^i \in R_n,$$

define the negacyclic convolution matrix

$$\mathrm{Rot}(a) \coloneqq \sum_{i=0}^{n-1} a[i]\, S^i \in \mathbb{Z}^{n\times n}.$$

Then, for all $a, b \in R_n$,

$$\mathrm{vec}(ab) = \mathrm{Rot}(a)\,\mathrm{vec}(b) = \mathrm{Rot}(b)\,\mathrm{vec}(a),$$

and $\mathrm{Rot}(\cdot)$ is a ring homomorphism:

$$\mathrm{Rot}(ab) = \mathrm{Rot}(a)\,\mathrm{Rot}(b), \qquad \mathrm{Rot}(1) = I_n.$$

Furthermore,

$$\mathrm{Rot}(a)^\top = \mathrm{Rot}(a^\star).$$

For

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in R_n^{2\times 2},$$

we define its lift to $\mathbb{Z}^{2n\times 2n}$ as

$$\mathrm{Lift}(M) \coloneqq \begin{pmatrix} \mathrm{Rot}(a) & \mathrm{Rot}(b) \\ \mathrm{Rot}(c) & \mathrm{Rot}(d) \end{pmatrix}.$$

For $\mathbf{u} = (u_0, u_1) \in R_n^2$, define

$$\mathrm{Vec}(\mathbf{u}) \coloneqq \big(\mathrm{vec}(u_0) \parallel \mathrm{vec}(u_1)\big) \in \mathbb{Z}^{2n}.$$

Then lifting linearizes ring multiplication:

$$\mathrm{Vec}(M\mathbf{u}) = \mathrm{Lift}(M)\,\mathrm{Vec}(\mathbf{u}), \qquad \mathrm{Lift}(MN) = \mathrm{Lift}(M)\,\mathrm{Lift}(N).$$

Thus bilinear forms over R_n^2 can be evaluated through integer linear algebra over $\mathbb{Z}^{2n}$.

Discrete Gaussians over parity cosets. For $t \in \{0,1\}$ and $\sigma > 0$, let $D_{2\mathbb{Z}+t,\sigma}$ denote the one-dimensional discrete Gaussian over the coset $2\mathbb{Z} + t$, namely

$$\Pr[X = 2k + t] \propto \exp\left(-\frac{(2k+t)^2}{2\sigma^2}\right), \qquad k \in \mathbb{Z}.$$

For a bit vector $t = (t_0, \ldots, t_{2n-1}) \in \{0,1\}^{2n}$, we write

$$D_{2\mathbb{Z}^{2n}+t,\sigma}$$

for the product distribution whose coordinates are sampled independently from the corresponding one-dimensional laws on $2\mathbb{Z} + t_i$. Hence

$$D_{2\mathbb{Z}^{2n}+t,\sigma}$$

is supported on the lattice coset $2\mathbb{Z}^{2n} + t$ and is spherically subgaussian. In particular, there exists an absolute constant $c > 0$ such that

$$\Pr\big[\|\mathbf{x}\|_2^2 > 2c(2n)\sigma^2\big] \le e^{-\Omega(n)}.$$

Therefore, bounds of the form

$$\|\mathbf{x}\|_2^2 \le 8n\,\sigma^2$$

hold except with negligible probability for the parameter ranges used in the scheme. When needed, we write

$$D_{2\mathbb{Z}^{2n}+t,\ 2\sigma_{\mathsf{sign}}}$$

to emphasize the signing parameter.

Each distribution $D_{2\mathbb{Z}+t,\sigma}$ is symmetric about zero. Accordingly, the reference sampler draws a nonnegative R_t according to the law of $|X|$ for $X \sim D_{2\mathbb{Z}+t,\sigma}$, independently draws $b_{\mathsf{sgn}} \sim \mathrm{Bernoulli}(1/2)$, and returns $\widetilde{X} = \varepsilon R_t$, where $\varepsilon \coloneqq (-1)^{b_{\mathsf{sgn}}} \in \{\pm 1\}$. Symmetry gives $\widetilde{X} \sim D_{2\mathbb{Z}+t,\sigma}$. The internal variable ε remains defined when $R_t = 0$.

For a single coordinate

$$X \sim D_{2\mathbb{Z}+t_i,\ 2\sigma_{\mathsf{sign}}},$$

we write

$$\mu_{\mathsf{sign}} \coloneqq \mathbb{E}[|X|]$$

for its expected absolute value. Its variance is

$$\mathrm{Var}(X) = 4\sigma_{\mathsf{sign}}^2.$$

2.2 The HAWK Scheme

We now summarize the algorithms of HAWK using the notation introduced in Section 2.1. The scheme consists of three algorithms: key generation, signing, and verification. For full details, we refer to the original HAWK specification [4].

Key generation (`HawkKeyGen`). The algorithm outputs a secret key

$$\mathsf{sk} = B \in \mathrm{GL}_2(R_n)$$

and a public key

$$\mathsf{pk} = \big(Q, \sigma_{\mathsf{sign}}, \sigma_{\mathsf{verify}}\big),$$

where

$$Q \coloneqq B^\star B$$

is a positive-definite Hermitian matrix with determinant 1.

To generate B, the algorithm first samples polynomials

$$f, g \in R_n$$

from a centered binomial distribution with parameter η, so that, in expectation,

$$\|f\|_2^2,\ \|g\|_2^2 \approx \frac{n\eta}{2}.$$

It then computes polynomials $F, G \in R_n$ satisfying the Bézout identity

$$fG - gF = 1,$$

and defines

$$B = \begin{pmatrix} f & F \\ g & G \end{pmatrix}.$$

The candidate is accepted only if (f, g) satisfy the required validity conditions, in particular that the constant coefficient of q_{00}^{-1} in $\mathbb{Q}[X]/(X^n + 1)$ is smaller than a prescribed bound β_0. Otherwise, the procedure restarts.

Signature generation (`HawkSign`). Given a message m and the public key pk, the signing algorithm outputs a signature

$$\sigma = (\mathsf{salt}, s_1).$$

(1) Compute

$$\mathrm{hpub} = \mathrm{SHAKE256}(\mathsf{pk}),$$
$$M = \mathrm{SHAKE256}(m \parallel \mathrm{hpub}),$$
$$h = (h_0, h_1) = \mathrm{SHAKE256}(M \parallel \mathsf{salt}) \in (R_n/2R_n)^2.$$

(2) Form

$$t \equiv Bh \pmod{2R_n}, \qquad t \in \{0,1\}^{2n},$$

and sample

$$\mathbf{x} = (x_0, x_1) \leftarrow D_{2\mathbb{Z}^{2n}+t,\ 2\sigma_{\mathsf{sign}}}.$$

(3) If

$$\|\mathbf{x}\|_2^2 > 8n\,\sigma_{\mathsf{verify}}^2,$$

restart with a fresh salt.

(4) Compute

$$\widehat{\mathbf{w}} = B^{-1}\mathbf{x} = h + 2\mathbf{s}, \qquad \mathbf{s} \in R_n^2.$$

In particular,

$$\widehat{w}_1 = -g\,x_0 + f\,x_1.$$

(5) Apply symmetry breaking. Let

$$i^\star = \min\{\, i \in \{0, \ldots, n-1\} : \widehat{w}_1[i] \neq 0 \,\},$$

and define

$$\varkappa = \mathrm{sgn}(\widehat{w}_1[i^\star]) \in \{\pm 1\}.$$

The canonical representative is then

$$\mathbf{w} \coloneqq \varkappa\,\widehat{\mathbf{w}},$$

so that the first nonzero coefficient of w_1 is positive.

(6) Output

$$\sigma = (\text{salt}, s_1), \qquad s_1 = \frac{h_1 - w_1}{2} \in R_n.$$

Signature verification (`HawkVerify`). Given a message m, the public key pk, and a signature

$$\sigma = (\text{salt}, s_1),$$

the verifier proceeds as follows.

(1) Recompute

$$\begin{aligned} \text{hpub} &= \text{SHAKE256}(\text{pk}), \\ M &= \text{SHAKE256}(m \,\|\, \text{hpub}), \\ h = (h_0, h_1) &= \text{SHAKE256}(M \,\|\, \text{salt}). \end{aligned}$$

(2) Reconstruct

$$\begin{aligned} w_1 &= h_1 - 2s_1, \\ s_0 &= \left\lceil \frac{h_0}{2} + \frac{q_{01}}{q_{00}} \left(\frac{h_1}{2} - s_1 \right) \right\rfloor, \\ w_0 &= h_0 - 2s_0, \end{aligned}$$

and define

$$\mathbf{w} = (w_0, w_1).$$

(3) Accept if and only if

$$\|\mathbf{w}\|_Q^2 = \mathrm{T}(\mathbf{w}^\star Q\, \mathbf{w}) \le 8n\, \sigma_{\text{verify}}^2.$$

3 Side-Channel Leakage in HAWK's DGS and the Induced Problem

The HAWK signature algorithm employs a DGS to generate signatures. This sampling procedure is particularly sensitive to side-channel leakage, meaning that observable quantities such as power consumption or electromagnetic emanations can reveal information about the internal state.

Such leakage can reveal partial information about the internal sample vectors, thereby making secret-key recovery more feasible. In this section, we review the leakage phenomena observed in HAWK's sampler and introduce a unified abstraction that we call *HAWK with Hint*.

3.1 Side-Channel Leakage in HAWK's DGS

```
1  #Inputs: tab_hi, tab_lo (RCDTs), v (index), t
       (target), hi, lo (randomness)
2  #Outputs: x[v] (sample at index v), sn (squared norm
       of x)
3
4  uint32_t neg = -(uint32_t)(lo >> 63);
5  uint32_t pbit = (t[v >> 3] >> (v & 7)) & 1;
6  uint64_t p_odd = -(uint64_t)pbit;
7  uint32_t p_oddw = (uint32_t)p_odd;
8  uint32_t r = 0;
9  for (size_t i = 0; i < hi_len; i += 2) {
10     uint64_t tlo0 = tab_lo[i + 0];
11     uint64_t tlo1 = tab_lo[i + 1];
12     uint64_t tlo = tlo0 ^ (p_odd & (tlo0 ^ tlo1));
13     uint32_t cc = (uint32_t)((lo - tlo) >> 63);
14     uint32_t thi0 = tab_hi[i + 0];
15     uint32_t thi1 = tab_hi[i + 1];
16     uint32_t thi = thi0 ^ (p_oddw & (thi0 ^ thi1));
17     r += (hi - thi - cc) >> 31;
18 }
19 uint32_t hinz = (hi - 1) >> 31;
20 for (size_t i = hi_len; i < lo_len; i += 2) {
21     uint64_t tlo0 = tab_lo[i + 0];
22     uint64_t tlo1 = tab_lo[i + 1];
23     uint64_t tlo = tlo0 ^ (p_odd & (tlo0 ^ tlo1));
24     uint32_t cc = (uint32_t)((lo - tlo) > > 63);
25     r += hinz & cc;
26 }
27 r = (r << 1) - p_oddw;
28 r = (r ^ neg) - neg;
29 x[v] = (int8_t)*(int32_t *)&r;
30 sn += r * r;
```

Listing 1: Discrete Gaussian Sampler inside the HAWK reference C implementation

The DGS invoked in HAWK's signing procedure performs a sequence of branching operations, table lookups, and arithmetic updates. Each of these steps produces characteristic side-channel leakage in power or electromagnetic traces, which can reveal information about the sampled coefficients. Following the analysis of [9], three types of leakage have been identified: sign, parity, and increment leakage. Listing 1 shows the process of sampling a single coefficient through the DGS in the HAWK reference C implementation. In Lines 4 and 5, the sign and parity bits are determined. From Lines 9 to 26, the sampled random value is compared with each element of the predefined tables, where the table selection is dictated by the previously obtained parity. In Lines 27 and 28, the determined sign and parity are applied, and finally, in Line 30, the norm is computed. In this code, the leakage points corresponding to the three types are as follows.

3.1.1 ***Sign Leakage.*** Sign information leaks during three distinct stages: (i) the initial sign decision (Line 4), (ii) the application of the chosen sign to the sampled coefficient (Line 28), and (iii) the norm computation used in the rejection step (Line 30). In practice, power traces separate cleanly into two groups, so that a simple threshold classifier suffices to distinguish the two values of the internal sign bit with essentially perfect accuracy. An additional detail is the treatment of zeros: in Lines 4 and 28, coefficients that are zero but marked with a "negative sign" produce a distinguishable trace (a so-called *negative zero*) [9]. Since the norm computation consistently treats zero as nonnegative in Line 30, these negative zeros can be reliably identified at that stage, further reinforcing the accuracy of sign recovery.

3.1.2 ***Parity Leakage.*** Parity information (whether a coefficient is even or odd) is determined early in the sampling routine (Line 5) and reappears multiple times during subsequent table traversals. In the specific implementation studied in [9], each iteration of the first traversal yields two leakage points (Lines 12 and 16), while the second traversal yields one (Line 23), resulting in a total of 18 parity-related leakage points per coefficient. These repeated exposures produce two distinct classes of traces, corresponding to even and odd values, with amplitudes dominated by Hamming-weight differences in 64-bit registers.

3.1.3 ***Increment Leakage.*** The magnitude of the sampled coefficient is determined through a sequential comparison procedure based on a Recursive Cumulative Distribution Table (RCDT). A uniformly random value is compared against table entries in descending order, and the coefficient is incremented whenever the random value falls below the current threshold. Each comparison operation produces a distinct side-channel footprint, allowing an

adversary to infer whether the magnitude was incremented at that iteration (Lines 13, 17). Although increment leakage is less robust than sign or parity leakage, it provides additional constraints on the distribution of coefficients and can be combined with other leakage to reduce the uncertainty about the sampled vector.

3.2 The HAWK with Hint Problem

The leakage described in Section 3.1 (sign, parity, and increment information) demonstrates that the DGS in HAWK reveals consistent, fine-grained clues about its internal outputs. We collectively refer to such side-channel–derived observations as *hints*. In our threat model, *HAWK with Hint* denotes the setting where an adversary obtains these clues in addition to the public transcript of the signing algorithm. The adversary's challenge is then twofold: (i) to extract hint information reliably from physical traces, and (ii) to leverage this information together with the public key to reconstruct the secret key.

The difficulty of this task depends critically on the form and quality of the hints, as well as on the number of signatures available. To formalize this problem, we adopt the following definition.

Definition 3.1. Let $B \in \mathrm{GL}_2(R_n)$ be the secret key of HAWK, and let $Q = B^\star B$ be the corresponding public key. For each signing execution j, the sampler produces vectors

$$(\boldsymbol{x}^{(j)}, \boldsymbol{w}^{(j)}) \in R_n^2 \times R_n^2$$

satisfying

$$\boldsymbol{x}^{(j)} = B\boldsymbol{w}^{(j)}, \qquad \boldsymbol{w}^{(j)} = h^{(j)} + 2s^{(j)}.$$

For the sign-leakage categories, let $\varepsilon_i^{(j)}[k] \in \{\pm 1\}$ denote the internal sign variable used when sampling $x_i^{(j)}[k]$, for $i \in \{0,1\}$ and $k \in \{0,\dots,n-1\}$. It remains defined when $x_i^{(j)}[k] = 0$. The adversary observes, in addition to the public transcript, side-channel information $I^{(j)}$.

Instance:

- The public key Q and transcript information from t signatures.
- Side-channel observations
$$\mathcal{I} = \{I^{(j)}\}_{j=1}^{t},$$
where each $I^{(j)}$ conveys side-channel information from the generation of $\mathrm{Vec}(\boldsymbol{x}^{(j)})$. In the sign-leakage categories, its coefficients are observations of $\varepsilon_i^{(j)}[k]$. The first n indices correspond to the coefficients of $x_0^{(j)}$, and the last n indices correspond to those of $x_1^{(j)}$.

Goal: Recover the secret basis B of HAWK.

At the physical level, HAWK's sampler exhibits sign, parity, and increment leakage. In our recovery analysis, however, we group these physical primitives into three coefficient-level regimes: full-coefficient, exact-sign, and noisy-sign leakage. We focus algorithmically on sign leakage because Section 5 shows that sign remains robust under noise whereas increment rapidly degrades. Depending on the leakage quality, we distinguish three categories:

- **Category I: Full coefficient recovery.** Each coefficient of $\boldsymbol{x}$ is recovered exactly, reducing key recovery to a direct linear-algebra problem.
- **Category II: Sign information.** Only the sampler's internal sign bit for each coefficient is obtained. This constrains each coefficient of $\boldsymbol{x}$ but does not uniquely determine it.
- **Category III: Noisy sign information.** The adversary obtains imperfect sign information, modeled in our analysis as independently corrupted sign bits. By combining these noisy observations with HAWK's algebraic structure, key recovery remains possible given sufficiently many signatures.

These categories abstract the spectrum of leakage scenarios for HAWK and provide the framework for the recovery algorithms developed in the next sections.

4 Key Recovery Analysis under HAWK with Hint

Unless stated otherwise, the proofs of all lemmas and theorems in this section are deferred to Appendix C.

4.1 Algebraic Preliminaries

We first summarize the algebraic identities that hold for any HAWK instance and that will be used throughout the key recovery analysis. Let

$$B = \begin{bmatrix} f & F \\ g & G \end{bmatrix} \in \mathrm{GL}_2(R_n), \qquad \det(B) = 1,$$

and define the public matrix

$$Q = B^\star B = \begin{bmatrix} q_{00} & q_{01} \\ q_{10} & q_{11} \end{bmatrix}.$$

Since $Q = B^\star B$, the matrix Q is the Gram matrix of the basis B, and is therefore Hermitian positive definite. In particular, it satisfies

$$BQ^{-1}B^\star = I_2.$$

Because Q is Hermitian, we have $q_{10} = q_{01}^\star$ and

$$q_{11} = \frac{|q_{01}|^2 + 1}{q_{00}}, \qquad |q_{01}|^2 = q_{01}^\star q_{10}. \tag{1}$$

During signing, for $\widehat{\mathbf{w}} \in R_n^2$ we compute

$$\mathbf{x} = B\widehat{\mathbf{w}}$$

and output

$$\mathbf{w} = \varkappa\,\widehat{\mathbf{w}}, \qquad \varkappa \in \{\pm 1\}.$$

Writing

$$\mathbf{x} = (x_0, x_1)^T, \qquad \mathbf{w} = (w_0, w_1)^T,$$

define

$$M_{\mathbf{x}} = \begin{bmatrix} x_0^\star & x_1^\star \\ -x_1 & x_0 \end{bmatrix},$$

$$M_{\mathbf{w}} = \begin{bmatrix} (\mathbf{w}^\star Q)_0 & (\mathbf{w}^\star Q)_1 \\ -w_1 & w_0 \end{bmatrix}, \tag{2}$$

$$M_{\widehat{\mathbf{w}}} = \begin{bmatrix} (\widehat{\mathbf{w}}^\star Q)_0 & (\widehat{\mathbf{w}}^\star Q)_1 \\ -\widehat{w}_1 & \widehat{w}_0 \end{bmatrix}.$$

Lemma 4.1. *For B, Q, $\mathbf{x}$, and $\widehat{\mathbf{w}}$ defined above, the following relations hold:*

$$M_{\widehat{\mathbf{w}}} B^{-1} = M_{\mathbf{x}}, \qquad M_{\widehat{\mathbf{w}}} = M_{\mathbf{x}} B. \tag{3}$$

Lemma 4.2. *For $\mathbf{w} = \varkappa\,\widehat{\mathbf{w}}$ with $\varkappa \in \{\pm 1\}$, the corresponding M-matrices satisfy*

$$M_{\mathbf{w}} = \varkappa\, M_{\widehat{\mathbf{w}}}. \tag{4}$$

Lemma 4.3. *The determinants of the matrices $M_{\mathbf{x}}$, $M_{\widehat{\mathbf{w}}}$, and $M_{\mathbf{w}}$ are equal and nonzero:*

$$\det(M_{\mathbf{w}}) = \det(M_{\widehat{\mathbf{w}}}) = \det(M_{\mathbf{x}}) = \mathbf{x}^{\star}\mathbf{x} \neq 0. \tag{5}$$

Lemmas 4.1–4.3 connect the observable signing vectors $(\mathbf{x}, \mathbf{w})$ to the hidden basis B and provide the algebraic foundation for the recovery algorithms developed in Sections 4.2–4.5.

We first study Category I in Section 4.2. Next, in Section 4.3, we address the `sym-break` issue that must be resolved for both Categories II and III. Finally, Sections 4.4 and 4.5 describe Categories II and III, respectively.

4.2 Category I

We begin with the strongest leakage model, Category I, in which the adversary observes both the sampled vector $\mathbf{x}$ and the corresponding canonical lattice vector $\mathbf{w}$ from a single signing execution. Guerreau and Rossi [9] required two exact pairs $(\mathbf{x}, \mathbf{w})$, with the corresponding public vectors forming an invertible matrix, whereas our recovery requires only one exact pair. Rahmani et al. [19] also obtain one-signature recovery under exact sampler-output leakage, but first recover one column of the secret basis and then reconstruct the remaining column from the public-key relations. Our Category I attack instead expresses the complete recovery through the unified matrix identity $M_{\mathbf{w}} = M_{\mathbf{x}}B$. Relative to Guerreau and Rossi, this eliminates the need to recover a second complete sampler output and removes the associated invertibility requirement, although it does not weaken the per-execution leakage assumption or improve the signature count over Rahmani et al. More importantly, the same identity provides the algebraic foundation for the constraint and feature constructions used with exact and noisy sign observations in Categories II and III. The recovery determines B only up to a global sign, yielding either B or $-B$, both of which induce the same public key $Q = B^{\star}B$.

Information inferred from leakage. In Category I, we assume that all coefficients of the sampled vector $\mathbf{x}$ are recovered exactly.

High-level idea. From the identities in Section 4.1, the observed pair $(\mathbf{x}, \mathbf{w})$ satisfies

$$M_{\mathbf{w}} = \varkappa M_{\widehat{\mathbf{w}}} = \varkappa M_{\mathbf{x}}B, \qquad \varkappa \in \{\pm 1\}.$$

Here $M_{\mathbf{x}}$ is computed directly from $\mathbf{x}$, and $M_{\mathbf{w}}$ is computed from $(\mathbf{w}, Q)$. By Lemma 4.3,

$$\det(M_{\mathbf{x}}) = \mathbf{x}^{\star}\mathbf{x} \neq 0.$$

Since K_n is a field, $M_{\mathbf{x}}$ is invertible over K_n. Therefore, we obtain the explicit candidate

$$\widetilde{B} := M_{\mathbf{x}}^{-1}M_{\mathbf{w}} = \varkappa B.$$

Although the inversion is carried out over K_n, the resulting matrix satisfies $\widetilde{B} = \varkappa B \in R_n^{2\times 2}$. Hence the secret basis is recovered up to a global sign using one matrix inversion and one matrix multiplication over K_n. This sign ambiguity is information-theoretically unavoidable from $(Q, \mathbf{x}, \mathbf{w})$ alone, since

$$(-B)^{\star}(-B) = B^{\star}B = Q.$$

Result. *Given a single observed pair $(\mathbf{x}, \mathbf{w})$ from one signing execution and the public key Q, the adversary can compute $\widetilde{B} = M_{\mathbf{x}}^{-1}M_{\mathbf{w}}$ and recover the secret basis up to sign, i.e., $\widetilde{B} \in \{B, -B\}$. This procedure runs in polynomial time in n.*

Theorem 4.4. *Let $B \in \mathrm{GL}_2(R_n)$ be the secret key of HAWK and let $Q = B^{\star}B$ be the public key. Suppose that for one signing execution the adversary obtains both $\mathbf{x} \in R_n^2$ and the canonical vector $\mathbf{w} \in R_n^2$, where*

$$\mathbf{w} = \varkappa\,\widehat{\mathbf{w}} \qquad \text{for some unknown } \varkappa \in \{\pm 1\},$$

and

$$\mathbf{x} = B\widehat{\mathbf{w}}.$$

Then the adversary can recover either B or $-B$ in polynomial time.

4.3 Handling `sym-break`

Before turning to Categories II and III, we first resolve the sample-wise sign symmetry introduced by HAWK's `sym-break`. Let $\mathbf{x}^{(j)}$ denote the sampled vector corresponding to the j-th signature. Each signature j involves an unknown sign

$$\varkappa^{(j)} \in \{\pm 1\}$$

that must be recovered before information can be aggregated across signatures. The spectral routine below aligns these signs and applies to both Category II (exact sign observations) and Category III (noisy sign observations).

Information inferred from leakage. For each signature $j \in \{1, \ldots, k\}$, the public transcript reveals the canonical signature vector

$$\mathbf{w}^{(j)} = (w_0^{(j)}, w_1^{(j)}) \in R_n^2$$

and the public key

$$Q = B^{\star}B \in R_n^{2\times 2}.$$

In addition, the side-channel attack yields partial information about the corresponding sampled vector

$$\mathbf{x}^{(j)} = (x_0^{(j)}, x_1^{(j)}) \in R_n^2.$$

We encode this leakage as

$$I^{(j)} = (I_0^{(j)}, I_1^{(j)}) \in R_n^2,$$

where each coefficient $I_i^{(j)}[k] \in \{\pm 1\}$ represents an observed sign and is corrupted according to an i.i.d. Bernoulli(p) flip variable, independently across coefficients and signatures:

$$I_i^{(j)}[k] = \begin{cases} \varepsilon_i^{(j)}[k] & \text{with probability } 1-p, \\ -\varepsilon_i^{(j)}[k] & \text{with probability } p. \end{cases}$$

We adopt this model as an analytically tractable baseline. Correlated or non-uniform classification errors are outside the scope of our guarantees.

Thus, the input to Algorithm 1 consists of the public vector $\mathbf{w}^{(j)}$, the side-channel sign pattern $I^{(j)}$, and the public key Q, while $\mathbf{x}^{(j)}$ itself remains hidden. Category II corresponds to the special case $p = 0$, while Category III allows $p > 0$.

The unknown symmetry-breaking sign $\varkappa^{(j)} \in \{\pm 1\}$ relates the canonical vector and the pre-canonical vector through

$$\mathbf{w}^{(j)} = \varkappa^{(j)}\widehat{\mathbf{w}}^{(j)}.$$

We now define the observable feature vector $\boldsymbol{\ell}^{(j)}$ and the corresponding theoretical feature vector $\mathbf{z}^{(j)}$.

Definition 4.5 (Feature vectors). Let $M_{\mathbf{w}}^{(j)}$ and $M_{\mathbf{x}}^{(j)}$ be the matrices defined in (2). For each signature j, define

$$\boldsymbol{\ell}^{(j)} := (M_{\mathbf{w}}^{(j)})^{\star} \begin{pmatrix} (I_0^{(j)})^{\star} \\ -I_1^{(j)} \end{pmatrix} \in R_n^2,$$

$$\mathbf{z}^{(j)} := (M_{\mathbf{x}}^{(j)})^{\star} \begin{pmatrix} (I_0^{(j)})^{\star} \\ -I_1^{(j)} \end{pmatrix} = \begin{pmatrix} \alpha^{(j)} \\ \beta^{(j)} \end{pmatrix} \in R_n^2.$$

Lemma 4.6. *Let $\boldsymbol{\ell}^{(j)}$ and $\mathbf{z}^{(j)}$ be as in Definition 4.5. Then the following hold.*

(1) ***Algebraic relation.*** *Using Lemmas 4.1 and 4.2,*

$$\boldsymbol{\ell}^{(j)} = \varkappa^{(j)} B^{\star} \mathbf{z}^{(j)}.$$

(2) ***Statistical moments.*** *The canonical embedding components are approximately distributed as*

$$\sigma(\alpha^{(j)})_k \sim \mathcal{CN}(2n\mu,\, 8n^2\sigma_{\text{sign}}^2), \qquad \sigma(\beta^{(j)})_k \sim \mathcal{CN}(0,\, 8n^2\sigma_{\text{sign}}^2)$$

for all $k \in \{0, \dots, n-1\}$, where

$$\mu = (1-2p)\mu_{\text{sign}}.$$

High-level idea. Our goal is to align the unknown symmetry-breaking signs so that information can be aggregated across signatures. To this end, we use the observable feature vectors $\{\boldsymbol{\ell}^{(j)}\}_{j=1}^k$ from Definition 4.5 and build a Gram matrix whose entries encode pairwise correlations under the Q^{-1} inner product. The following lemma records the resulting spike structure, which is the key to the spectral sign-recovery procedure.

Lemma 4.7. *Let $G_0 \in \mathbb{R}^{k\times k}$ have entries*

$$(G_0)_{ij} := \mathrm{T}\big((\boldsymbol{\ell}^{(i)})^{\star} Q^{-1} \boldsymbol{\ell}^{(j)}\big),$$

where $\boldsymbol{\ell}^{(j)}$ and $\mathbf{z}^{(j)}$ are as in Definition 4.5. Then

$$(G_0)_{ij} = \varkappa^{(i)}\varkappa^{(j)}\, \mathrm{T}\big((\mathbf{z}^{(i)})^{\star}\mathbf{z}^{(j)}\big).$$

In particular, under the noisy-sign model of Lemma 4.6 with flip rate $p < \frac{1}{2}$,

$$\mathbb{E}[G_0] = \sigma_0 I_k + \lambda\, \psi\psi^{\top}, \qquad \psi = (\varkappa^{(1)}, \dots, \varkappa^{(k)})^{\mathsf{T}},$$

with

$$\lambda = 4n^2\mu^2, \qquad \sigma_0 = 16n^2\sigma_{\text{sign}}^2,$$

where $\mu = (1-2p)\mu_{\text{sign}}$.

Intuitively, $\mathbb{E}[G_0]$ consists of an isotropic noise component and a rank-one spike in the direction of the unknown sign vector ψ. Therefore, the top eigenvector of G_0 is expected to align with ψ.

Algorithm. Algorithm 1 summarizes the resulting spectral method for recovering the symmetry-breaking signs. The procedure constructs observable feature vectors, forms the Gram matrix implicitly via the lifted Q^{-1}, and extracts the signs from the leading eigenvector.

We express

$$G_0 = A\,\mathrm{Lift}(Q^{-1})A^{\top}$$

with

$$A \in \mathbb{R}^{k\times 2n}, \qquad \mathrm{Lift}(Q^{-1}) \in \mathbb{R}^{2n\times 2n},$$

Algorithm 1: Symmetry-Breaking Sign Recovery via Spectral Method

Input: $\{(\mathbf{w}^{(j)}, I^{(j)})\}_{j=1}^k$ where $\mathbf{w}^{(j)} \in R_n^2$, $I^{(j)} = (I_0^{(j)}, I_1^{(j)}) \in R_n^2$ with coefficients in $\{\pm 1\}$, public key $Q \in R_n^{2\times 2}$

Output: Symmetry-breaking signs $\{\varkappa^{(j)}\}_{j=1}^k$, aligned vectors $\{\widehat{\mathbf{w}}^{(j)}\}_{j=1}^k$, and feature vectors $\{\boldsymbol{\ell}^{(j)}\}_{j=1}^k$

for $j \leftarrow 1$ **to** k **do**
  Compute $M_{\mathbf{w}}^{(j)}$ from $\mathbf{w}^{(j)}$ and Q via (2)
  $\boldsymbol{\ell}^{(j)} \leftarrow (M_{\mathbf{w}}^{(j)})^{\star} \begin{pmatrix} (I_0^{(j)})^{\star} \\ -I_1^{(j)} \end{pmatrix}$
  $A_j \leftarrow \mathrm{vec}(\boldsymbol{\ell}^{(j)}) \in \mathbb{R}^{2n}$
$A \leftarrow [A_1; \dots; A_k] \in \mathbb{R}^{k\times 2n}$
Compute the leading eigenvector $\widehat{\psi} \in \mathbb{R}^k$ of $G_0 = A\,\mathrm{Lift}(Q^{-1})A^{\top}$ using the Lanczos method without forming G_0 explicitly
for $j \leftarrow 1$ **to** k **do**
  $\varkappa^{(j)} \leftarrow \mathrm{sign}(\widehat{\psi}_j)$
  $\widehat{\mathbf{w}}^{(j)} \leftarrow \varkappa^{(j)}\mathbf{w}^{(j)}$
return $\{\varkappa^{(j)}\}_{j=1}^k$, $\{\widehat{\mathbf{w}}^{(j)}\}_{j=1}^k$, *and* $\{\boldsymbol{\ell}^{(j)}\}_{j=1}^k$

thereby avoiding explicit construction of the $k \times k$ matrix. The Lanczos method then computes the top eigenvector in time $O(Tkn)$, where T is the number of iterations, so the overall complexity is polynomial in both n and k.

Result. We now state the performance guarantee for the spectral sign-alignment procedure. It follows from Lemma 4.7, standard matrix concentration bounds, and the Davis–Kahan theorem.

Theorem 4.8. *Given k pairs $\{(\mathbf{w}^{(j)}, I^{(j)})\}_{j=1}^k$ whose signs follow the noisy-sign model, if*

$$k \geq \frac{8\sigma_{\text{sign}}^2 (\log k)^2}{n\mu^2}\left(1 + \frac{2\sigma_{\text{sign}}^2}{\mu^2}\right),$$

where $\mu = (1-2p)\mu_{\text{sign}}$, then the algorithm runs in polynomial time

$$T_1 = O(Tkn)$$

and, for each $j = 1, \dots, k$, with probability $1 - O(1/k)$, the output of Algorithm 1 satisfies

$$\mathrm{sign}(\widehat{\psi}_j) = \varkappa^{(j)}.$$

This covers both Categories II and III.

4.4 Category II

Information inferred from leakage. After the symmetry-breaking step in Algorithm 1, we assume that the pre-canonical vectors $\{\widehat{\mathbf{w}}^{(j)}\}_{j=1}^k$ have been correctly aligned. For each signature j, the public transcript reveals the canonical vector $\mathbf{w}^{(j)} \in R_n^2$ and the public key $Q = B^{\star}B \in R_n^{2\times 2}$, while the side-channel leakage provides the sign information of the corresponding sampled vector $\mathbf{x}^{(j)} \in R_n^2$.

In Category II, this leakage contains only exact signs of all coefficients, encoded as

$$I^{(j)} = (I_0^{(j)}, I_1^{(j)}) \in R_n^2$$

with

$$I_i^{(j)}[k] = \varepsilon_i^{(j)}[k] \in \{\pm 1\}.$$

This is precisely the noiseless case $p = 0$ of the noisy-sign model from Section 4.3.

High-level idea. From the aligned pre-canonical vectors $\widehat{\mathbf{w}}^{(j)}$ and the algebraic relation in Lemma 4.1, each observed sign pattern yields linear constraints on

$$\mathbf{z}_{\text{key}} = \text{Vec}(G, -g),$$

which is the first column of B^{-1}. Crucially, these constraints correspond to absolute values of coefficients sampled from a discrete Gaussian and are therefore nonnegative integers.

Geometrically, the constraints define parallel hyperplanes orthogonal to $\mathbf{z}_{\text{key}}$, indexed by $k \in \{0, 1, 2, \ldots\}$. Among them, the zero-coefficient constraints lie exactly on the hyperplane $k = 0$ through the origin. When sufficiently many such constraints are in general position inside the orthogonal subspace, the feasible cone collapses to a single ray in the direction of $\mathbf{z}_{\text{key}}$, thereby uniquely determining the secret basis.

We formalize this structure below and then state the main recovery theorem.

Definition 4.9. For each signature $j \in \{1, \ldots, k\}$, let

$$\mathbf{v}^{(j)} = \text{Vec}\big((I_0^{(j)})^\star, -I_1^{(j)}\big) \in \{\pm 1\}^{2n}$$

encode the observed sign pattern. Define the constraint matrix

$$A^{(j)} = \text{Diag}(\mathbf{v}^{(j)}) \cdot \text{Lift}\big(M_{\widehat{\mathbf{w}}^{(j)}}\big) \in \mathbb{R}^{2n \times 2n}.$$

The aggregate constraint matrix

$$A \in \mathbb{R}^{m \times 2n}, \qquad m = 2nk,$$

is obtained by stacking all constraint matrices vertically:

$$A = \begin{bmatrix} A^{(1)} \\ \vdots \\ A^{(k)} \end{bmatrix}.$$

We denote the rows of A by $\mathbf{a}_i \in \mathbb{R}^{2n}$ for $i = 1, \ldots, m$.

Property 4.10. Let

$$\mathbf{z}_{\text{key}} = \text{Vec}(G, -g) \in \mathbb{Z}^{2n}$$

denote the first column of B^{-1}. Then

$$A\mathbf{z}_{\text{key}} \succeq \mathbf{0}.$$

Moreover, each entry

$$(A\mathbf{z}_{\text{key}})_i = \langle \mathbf{a}_i, \mathbf{z}_{\text{key}} \rangle$$

equals the absolute value of a coefficient of some $\mathbf{x}^{(j)}$ and is therefore a nonnegative integer.

Algorithm 2: Category II: Key Recovery via Linear Program

Input: Public key $Q = B^\star B \in R_n^{2\times 2}$, aligned pre-canonical vectors $\{\widehat{\mathbf{w}}^{(j)}\}_{j=1}^k$ and exact sign patterns $\{I^{(j)}\}_{j=1}^k$ from Algorithm 1

Output: Secret basis B up to a global sign

for $j = 1$ **to** k **do**
- Compute $M_{\widehat{\mathbf{w}}}^{(j)}$ from $\widehat{\mathbf{w}}^{(j)}$ and Q via (2)
- Compute sign vector $\mathbf{v}^{(j)} \leftarrow \text{Vec}\big((I_0^{(j)})^\star, -I_1^{(j)}\big)$
- Build constraint matrix $A^{(j)} \leftarrow \text{Diag}(\mathbf{v}^{(j)}) \cdot \text{Lift}\big(M_{\widehat{\mathbf{w}}}^{(j)}\big)$

Stack all constraints into $A \in \mathbb{R}^{m\times 2n}$ with $m = 2nk$

$\mathbf{z}^* \leftarrow \text{IPM}(A)$

$\text{sol}_{\min} \leftarrow \min\{|\mathbf{z}_i^*| : |\mathbf{z}_i^*| > \varepsilon\}$

$\mathbf{z}_{\text{key}} \leftarrow \text{round}(\mathbf{z}^*/\text{sol}_{\min})$

Write $\mathbf{z}_{\text{key}} = \text{Vec}(G, -g)$

$(f, F) \leftarrow (G^\star, -g^\star)\, Q$

return B

To find a feasible solution satisfying

$$A\mathbf{z} \succeq \mathbf{0},$$

we convert the constraint system into a linear program and solve it using an interior-point method. The full LP formulation, including the objective, normalization constraints, and implementation details, is given in Appendix D.

Algorithm. The recovery proceeds in two stages. First, we obtain $\widehat{\mathbf{w}}^{(j)}$ for each signature via the symmetry-breaking algorithm (Algorithm 1). Second, we construct the inequalities

$$A^{(j)}\mathbf{z}_{\text{key}} \succeq \mathbf{0}$$

from the aligned pre-canonical vectors and sign patterns and solve the corresponding linear program using an interior-point method to recover the ray whose direction equals $\mathbf{z}_{\text{key}}$, from which the basis B is reconstructed.

The naive formulation has complexity $O(k^3n^3)$ because it solves a system of dimension $(2n + 2kn + 1) \times (2n + 2kn + 1)$. By exploiting the structure of the weight matrix in the interior-point iterations, the system can be reduced to dimension $(2n + 1) \times (2n + 1)$, yielding complexity $O(kn^3)$. A detailed description is provided in Appendix D.

Result. We now state the performance guarantee for the linear-programming approach in Category II. The success probability depends on the number of zero-coefficient constraints being in general position, which we analyze using Wendel's theorem on random hyperplane arrangements.

Theorem 4.11. *Let B be the secret key and $Q = B^\star B$ the public key of HAWK. Given k pairs $\{(\widehat{\mathbf{w}}^{(j)}, I^{(j)})\}_{j=1}^k$ with exact sign observations ($p = 0$), the secret basis can be recovered in time*

$$T_2 = O(kn^3),$$

and the success probability is

$$\Pr[\text{key recovery} \mid k] = \mathbb{E}_{m\sim \text{Bin}(2nk, p_0)}[P_{2n-1}(m)],$$

where

$$p_0 = \tfrac{1}{2} \Pr_{Z \sim D_{\mathbb{Z},\sigma_{\text{sign}}}} [Z = 0]$$

and

$$P_d(m) := 1 - 2^{-m+1} \sum_{j=0}^{d-1} \binom{m-1}{j}$$

is the probability that m random hyperplanes through the origin in $\mathbb{R}^d$ *are in general position [22].*

For exact sign observations ($p = 0$), Theorem 4.8 guarantees successful symmetry-breaking sign recovery.

Moreover, $P_d(m)$ converges rapidly to 1 once $m \geq 2d$. For the relevant dimension $d = 2n - 1$, recovery becomes highly likely once the number of zero-coefficient constraints exceeds roughly $2(2n-1)$. Since

$$m \sim \text{Bin}(2kn, p_0) \qquad \text{and} \qquad \mathbb{E}[m] = 2knp_0,$$

key recovery is therefore expected when

$$k \gtrsim 2/p_0.$$

The total time complexity combines spectral alignment and linear programming, yielding

$$T_1 + T_2 = O(Tkn + kn^3).$$

4.5 Category III

Information inferred from leakage. Following the symmetry-breaking step in Algorithm 1, we assume that the pre-canonical vectors $\{\widehat{\mathbf{w}}^{(j)}\}_{j=1}^k$ are correctly aligned. In Category III, the side-channel leakage provides noisy sign information about the sampled vectors

$$\mathbf{x}^{(j)} \in R_n^2.$$

For each signature j, the public transcript reveals $\mathbf{w}^{(j)} \in R_n^2$ and $Q = B^\star B$, while the leakage is represented as

$$I^{(j)} = (I_0^{(j)}, I_1^{(j)}) \in R_n^2$$

with coefficients $I_i^{(j)}[k] \in \{\pm 1\}$ flipped independently with probability $p < 1/2$:

$$I_i^{(j)}[k] = \begin{cases} \varepsilon_i^{(j)}[k] & \text{with probability } 1-p, \\ -\varepsilon_i^{(j)}[k] & \text{with probability } p. \end{cases}$$

This strictly generalizes the noiseless setting $p = 0$ of Category II.

High-level idea. From the aligned pre-canonical vectors $\widehat{\mathbf{w}}^{(j)}$ obtained in Section 4.3, we have recovered the symmetry-breaking signs $\{\varkappa^{(j)}\}_{j=1}^k$. By the algebraic relation in Lemma 4.6, each signature yields

$$\varkappa^{(j)} \boldsymbol{\ell}^{(j)} = B^\star \mathbf{z}^{(j)},$$

where $\boldsymbol{\ell}^{(j)}$ is computable from public information and

$$\mathbf{z}^{(j)} = \begin{pmatrix} \alpha^{(j)} \\ \beta^{(j)} \end{pmatrix}$$

has moments

$$\mathbb{E}[\alpha^{(j)}] = c, \qquad \mathbb{E}[\beta^{(j)}] = 0, \qquad c = \Theta(n(1-2p)).$$

The key idea is that if we sum these vectors over all signatures and normalize, the aggregate concentrates near the direction $(1,0)^T$:

$$\frac{\mathbf{z}_\Sigma}{\|\mathbf{z}_\Sigma\|} \approx \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad \mathbf{z}_\Sigma = \sum_{j=1}^k \mathbf{z}^{(j)}.$$

This reveals the first column of $B^\star$, because

$$\frac{\boldsymbol{\ell}_\Sigma}{\|\mathbf{z}_\Sigma\|} = B^\star \frac{\mathbf{z}_\Sigma}{\|\mathbf{z}_\Sigma\|} \approx B^\star \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} f^\star \\ F^\star \end{pmatrix}.$$

Crucially, the normalization factor $\|\mathbf{z}_\Sigma\|^2$ can be computed from public information through

$$(\boldsymbol{\ell}_\Sigma)^\star Q^{-1} \boldsymbol{\ell}_\Sigma = (\mathbf{z}_\Sigma)^\star \mathbf{z}_\Sigma,$$

which enables slot-wise normalization in the canonical embedding followed by coefficient-wise rounding.

We formalize the aggregation procedure below.

Definition 4.12. For signatures $j \in \{1, \ldots, k\}$ with aligned symmetry-breaking signs $\{\varkappa^{(j)}\}_{j=1}^k$ and feature vectors $\{\boldsymbol{\ell}^{(j)}, \mathbf{z}^{(j)}\}_{j=1}^k$ from Definition 4.5, define the aligned aggregate feature

$$\boldsymbol{\ell}_\Sigma := \sum_{j=1}^k \varkappa^{(j)} \boldsymbol{\ell}^{(j)} \in R_n^2$$

and the theoretical aggregate

$$\mathbf{z}_\Sigma := \sum_{j=1}^k \mathbf{z}^{(j)} = \begin{pmatrix} \alpha_\Sigma \\ \beta_\Sigma \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^k \alpha^{(j)} \\ \sum_{j=1}^k \beta^{(j)} \end{pmatrix} \in R_n^2.$$

Lemma 4.13. *The aggregate vectors from Definition 4.12 satisfy the following properties.*

(1) ***Algebraic relations.*** *Using Lemma 4.6 and the Gram relation* $BQ^{-1}B^\star = I_2$*, we have*

$$\boldsymbol{\ell}_\Sigma = B^\star \mathbf{z}_\Sigma \qquad \text{and} \qquad (\boldsymbol{\ell}_\Sigma)^\star Q^{-1} \boldsymbol{\ell}_\Sigma = (\mathbf{z}_\Sigma)^\star \mathbf{z}_\Sigma.$$

(2) ***Normalization property.*** *Under the noisy-sign model with independent flip rate* $p < \frac{1}{2}$*, the normalized aggregate*

$$\mathbf{z}_\Sigma / \|\mathbf{z}_\Sigma\|$$

concentrates near the direction $(1,0)^T$ *with per-component error variance* $\Theta(1/k)$.

Algorithm. The recovery again proceeds in two stages. First, we apply Algorithm 1 to recover the symmetry-breaking signs $\{\varkappa^{(j)}\}_{j=1}^k$. Second, we aggregate the aligned features

$$\boldsymbol{\ell}_\Sigma = \sum_{j=1}^k \varkappa^{(j)} \boldsymbol{\ell}^{(j)},$$

compute the normalization factor through

$$(\boldsymbol{\ell}_\Sigma)^\star Q^{-1} \boldsymbol{\ell}_\Sigma = (\mathbf{z}_\Sigma)^\star \mathbf{z}_\Sigma,$$

perform slot-wise normalization in the canonical embedding to approximate $f^\star(\zeta^s)$, and recover f by coefficient-wise rounding, from which the full basis B is reconstructed.

The complexity is $O(Tkn)$ for Stage 1, where T is the number of Lanczos iterations, and $O(kn + n \log n)$ for Stage 2, yielding overall polynomial-time complexity in both n and k. A concrete description is given in Algorithm 3.

Algorithm 3: Category III: Key Recovery from Noisy Sign Information

Input: Public key $Q = B^\star B \in R_n^{2\times 2}$, symmetry-breaking signs $\{\varkappa^{(j)}\}_{j=1}^k \subset \{\pm 1\}$ and feature vectors $\{\boldsymbol{\ell}^{(j)}\}_{j=1}^k$ from Algorithm 1

Output: Secret basis B up to a global sign

$\boldsymbol{\ell}_\Sigma \leftarrow \mathbf{0} \in R_n^2$
for $j \leftarrow 1$ **to** k **do**
  $\boldsymbol{\ell}_\Sigma \leftarrow \boldsymbol{\ell}_\Sigma + \varkappa^{(j)}\boldsymbol{\ell}^{(j)}$
$T \leftarrow (\boldsymbol{\ell}_\Sigma)^\star Q^{-1}\boldsymbol{\ell}_\Sigma \in R_n$
for *each embedding slot* $s \in \{1, 3, \ldots, 2n-1\}$ **do**
  $r_s \leftarrow \sqrt{T(\zeta^s)} \in \mathbb{R}_{>0}$
  $\widehat{f}(\zeta^s) \leftarrow \big(\boldsymbol{\ell}_\Sigma^{(0)}\big)^\star(\zeta^s)/r_s \in \mathbb{C}$
Apply the inverse canonical embedding to obtain $\widehat{f} \in K_n$
$f \leftarrow$ round each coefficient of $\widehat{f}$ to the nearest integer
Recover F, g, and G from f using the relations $Q = B^\star B$
return B

Result. Combining eigenvector concentration, Gaussian tail bounds, and the moment relations above yields the following guarantee.

Theorem 4.14. *Suppose the symmetry-breaking signs* $\{\varkappa^{(j)}\}_{j=1}^k$ *have been correctly recovered. If*

$$k \geq \frac{12C\eta\sigma_{\text{sign}}^2 \log n}{\mu^2},$$

where $\mu = (1-2p)\mu_{\text{sign}}$*, then the aggregation and slot-wise normalization procedure recovers* $f^\star$ *(and hence the secret basis* B*) in time*

$$T_3 = O(kn + n\log n)$$

via coefficient-wise rounding, with probability at least

$$1 - n^{1-\Omega(C)},$$

where $C > 0$ *is an absolute constant controlling the confidence level.*

Together with Theorem 4.8, this implies that whenever k simultaneously satisfies the spectral-alignment condition and the lower bound in Theorem 4.14, the two-stage Category III procedure recovers B in time

$$T_1 + T_3 = O(Tkn + kn + n\log n)$$

with overwhelming probability.

5 Measured Leakage and Recovery Evaluation

We first characterize the extractability and noise sensitivity of individual leakage components using isolated-coefficient traces. We then demonstrate HAWK-1024 key recovery using sign leakage measured from complete `sig_gauss()` executions within actual signing operations. Finally, experiments under controlled leakage models evaluate the recovery algorithms across parameter sets, signature counts, and sign-error rates.

5.1 Isolated-Coefficient Sampler-Leakage Characterization

5.1.1 Experimental setup. All power traces were collected using the ChipWhisperer–Husky platform targeting an ARM Cortex-M4–based STM32F3 microcontroller. Following the methodology of prior work [9], the `SamplerSign` code was modified to sample a single coefficient by isolating the corresponding loop in the HAWK-1024 reference implementation. Each execution that sampled one coefficient was recorded as a single power trace. We collected 100 such traces as the baseline dataset for the controlled noise experiments.

5.1.2 Controlled noise injection. To evaluate leakage degradation under a reproducible range of measurement conditions, we injected artificial Gaussian noise into the collected baseline traces. We estimate σ_{trace} as a scalar baseline noise level by computing the per-sample standard deviation across repeated acquisitions, selecting background windows unrelated to sampling activity, and averaging the standard deviations within those windows. These windows correspond to regions not directly involved in the sampling process, and the measured standard deviations are assumed to reflect the background measurement noise.

Based on σ_{trace}, we set the standard deviation of the injected noise, and for a chosen scaling factor α, the noisy trace at time index i is modeled as:

$$\text{Trace}_{\text{noisy}}[i] = \text{Trace}[i] + \mathcal{N}\big(0, (\alpha \cdot \sigma_{\text{trace}})^2\big). \tag{6}$$

In this paper, we considered scaling factors $\alpha \in \{1, 3, 5, 7, 10\}$ to systematically evaluate side-channel leakage variations under different noise levels. For each value of α, we generated 1,000 independent noise realizations from the same 100 baseline traces. Within each realization, Gaussian noise was independently added to every baseline trace, yielding a set of $T = 100$ noisy traces.

5.1.3 Information extraction procedure. Our side-channel analysis pipeline consists of the following stages:

(1) **Input trace set:** Each noise realization consists of $T = 100$ noisy traces, one derived from each baseline trace.
(2) **Preprocessing and alignment:** Because each trace targets an isolated single-coefficient sampling operation, no additional alignment is required in this experiment. The segmentation required for the complete `sig_gauss()` execution is treated separately in Section 5.2.1.
(3) **Feature extraction:** To extract sign, parity, and increment information from the power traces, it is necessary to identify the points of interest (PoI) where such leakages occur. PoIs can be determined based on the structural characteristics of the HAWK implementation, for example, the intervals of repeated table comparisons as illustrated in Listing 1, the locations within each iteration that exhibit high variance across trace indices, and regions identified through SNR analysis. In our experiment, PoI identification was performed directly on the raw traces.
(4) **Clustering:** For each realization, k-means clustering with $k = 2$ was applied to the $T = 100$ leakage values at each PoI to extract the sign, parity, and increment information.

Table 1: Average accuracy of sign, parity, and increment leakage under different noise levels (1,000 trials).

α	Sign (acc.)	Parity (acc.)	Increment (acc.)
Original	100.0%	100.0%	100.0%
1	100.0%	100.0%	99.3%
3	99.9%	100.0%	89.8%
5	98.8%	99.9%	79.0%
7	96.8%	98.4%	72.2%
10	92.3%	93.2%	66.2%

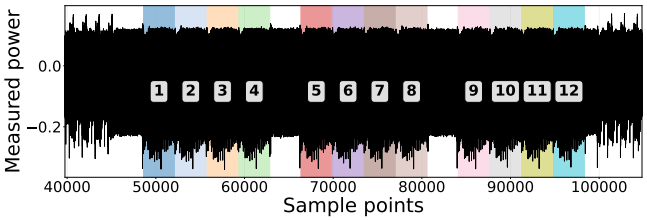


Figure 1: Template-based segmentation of a measured HAWK-1024 `sig_gauss()` trace. The colored regions indicate 12 consecutive coefficient-sampler windows.

(5) **Evaluation:** The clustering results obtained at each sign, parity, and increment leakage point were compared with the ground-truth labels to compute the accuracy for that realization.

5.1.4 Information extraction accuracy. During HAWK coefficient sampling, we evaluated the leakage components corresponding to the hints used in our recovery formulation. The sign, parity, and increment components reflect different coefficient properties in the power traces and respond differently to injected noise. Representative leakage traces across various noise levels are presented in Appendix E.

Sign and parity leakages form two clearly separated clusters that distinguish negative from positive coefficients and odd from even coefficients. These clusters remain visually distinct as the injected noise increases. In contrast, increment leakage, which reflects whether an internal table comparison increases the coefficient value, is much weaker. Because the Hamming-weight difference between increment and non-increment cases is relatively small, the separation between the two clusters collapses rapidly as noise increases.

Table 1 presents the average extraction accuracy for each leakage type as the injected-noise parameter α varies. Following the clustering-based approach of prior work [9], we apply k-means clustering to the extracted leakage signals and average the extraction accuracies over 1,000 independent noise realizations for each noise level. Without artificially injected noise, all three leakage types are extracted with 100% accuracy. As the noise level increases, sign and parity accuracies remain nearly perfect at low and moderate noise levels and remain approximately 93% at the highest tested level. Increment accuracy deteriorates much faster, reaching approximately two-thirds at $\alpha = 10$.

The robustness of sign leakage observed above motivates the design of Categories II and III, both of which rely exclusively on sign information.

5.2 Sampler-Stage End-to-End Key Recovery on HAWK-1024

We evaluate whether a complete `sig_gauss()` execution can be segmented into its $2n$ coefficient-sampler invocations and whether the extracted sign information is sufficient for HAWK-1024 key recovery.

5.2.1 Trace acquisition and sampler-window segmentation. We place the acquisition trigger around the complete execution of `sig_gauss()` during signing. Each trace therefore contains the $2n = 2{,}048$ sampler invocations that generate the two components of the HAWK-1024 sample vector. The ChipWhisperer–Husky was configured with an ADC clock multiplier of 2, a gain of 20 dB, and streaming acquisition of 14,477,416 sample points per trace. We collected 14 successful signing traces under one fixed key using independently generated 16-byte messages. Rejected signing executions were excluded.

We locate the sampler invocations using template matching. As shown in Listing 1, each coefficient follows the same operation sequence, producing a repeatable temporal pattern despite value-dependent amplitude variations. We select a representative 3,480-sample coefficient interval as the template and compute its normalized cross-correlation over the complete trace. Starting from the highest scores, we greedily select nonoverlapping candidates separated by at least 250 sample points until $2n$ windows are obtained. The selected windows are then sorted temporally and aligned at their template start positions. This procedure identified all 2,048 sampler windows in each of the 14 traces.

Figure 1 shows 12 consecutive windows in the average of five successful signing traces. The windows form three groups of four because the implementation obtains additional SHAKE256 output after every four coefficient samplings. The selected matches have normalized cross-correlation scores between 0.948 and 0.957, with a median of 0.954.

5.2.2 Sign extraction from measured traces. Guided by the sign-leakage points identified during the isolated-coefficient characterization, we compute the point-wise variance across the 2,048 aligned windows for each signature. We identify a prominent peak at offset 3,456 and use the amplitude at this offset as the sign-leakage feature.

The 2,048 leakage values from each signature are partitioned using one-dimensional k-means clustering with $k = 2$. The cluster with the larger centroid is mapped to +1, and the other is mapped to −1. We then apply a fixed, data-independent permutation derived from the reference implementation to convert the temporal sampling order into the coefficient order required by the recovery algorithm. The first n entries correspond to x_0, and the remaining n entries correspond to x_1. This produces $I^{(j)} \in \{\pm 1\}^{2n}$ for each signature j.

Table 2: HAWK parameter sets and derived statistical quantities.

Parameter Set	n	η	σ_{sign}	μ_{sign}	$\sigma^2_{\text{sign}}/\mu^2_{\text{sign}}$	p_0
HAWK-256	256	2	1.010	1.578	0.409	0.197
HAWK-512	512	4	1.278	2.013	0.403	0.156
HAWK-1024	1024	8	1.299	2.047	0.403	0.154

Comparison with instrumented ground-truth labels showed zero sign errors across all 14 signatures and 28,672 coefficient observations. These labels were used only to evaluate extraction accuracy and were not provided to the recovery algorithm.

5.2.3 Key recovery and verification. For each measured signature j, the recovery algorithm receives only the public key Q, the corresponding public signature transcript, and the extracted sign vector $I^{(j)}$. We align the observations using the symmetry-breaking procedure from Section 4.3 and then apply the Category II linear-programming attack from Section 4.4.

The recovered candidate $\widehat{B}$ is verified by checking

$$\widehat{B}^\star \widehat{B} = Q.$$

Recovery of either B or $-B$ is considered successful because both induce the same public key.

Using all 14 measured signatures, the sign-only pipeline successfully recovered the fixed HAWK-1024 key. This experiment demonstrates end-to-end measured key recovery for one key but does not estimate the success probability over independently generated keys. Multi-key success rates under controlled leakage models are evaluated in the following subsection.

5.3 Recovery evaluation under controlled sign-leakage models

We complement the physical experiments with controlled simulations for HAWK-256, HAWK-512, and HAWK-1024. These simulations bypass trace segmentation and classification: Category II receives exact sign observations, whereas Category III receives sign observations corrupted by simulated i.i.d. Bernoulli(p) flips across coefficients and signatures. This separation allows us to evaluate the recovery algorithms over a wider range of dimensions and error rates than the measured HAWK-1024 experiment alone.

Since Category I (full coefficient leakage) achieves deterministic key recovery from a single signature, we focus the statistical evaluation on Categories II and III. The recovery algorithms were implemented in Python using NumPy/SciPy and executed on an AWS EC2 instance. Unless otherwise noted, each reported setting is evaluated over 100 independent trials, and every trial uses an independently generated HAWK key pair. Table 2 summarizes the three HAWK parameter sets and their derived statistical quantities, where $\mu_{\text{sign}} = \frac{1}{2}\left(\mathbb{E}[\|X_0\|] + \mathbb{E}[\|X_1\|]\right)$ for coset Gaussians $X_i \sim D_{2\mathbb{Z}+i,\,2\sigma_{\text{sign}}}$, and $p_0 = \frac{1}{2}\Pr[Z = 0]$ for $Z \sim D_{\mathbb{Z},\sigma_{\text{sign}}}$ represents the zero-coefficient probability.

5.3.1 Category II: Exact sign leakage. Category II attacks assume the adversary obtains exact sign information for each coefficient of $\mathbf{x}^{(j)}$ across k signatures. Table 3 shows the empirical success rates as k varies. HAWK-256 achieves 100% empirical success over 100 trials using $k = 11$ signatures, whereas HAWK-512 and HAWK-1024 achieve 100% empirical success using $k = 14$ signatures. Empirical success rates consistently exceed theoretical predictions across all parameter sets: for instance, HAWK-256 achieves 56% at $k = 10$ versus the 39.8% theoretical prediction. This systematic improvement may arise from the negacyclic structure of HAWK's polynomial ring $\mathbb{Z}[X]/(X^n + 1)$, where algebraic dependencies can cause constraint vectors to exhibit greater angular spread than the random-sphere model predicts.

Table 3: Category II success rates over 100 trials.

	Signatures k					
Parameter	10	11	12	13	14	15
HAWK-256	56%	**100%**	100%	100%	100%	100%
HAWK-512	0%	0%	1%	75%	**100%**	100%
HAWK-1024	0%	0%	0%	30%	**100%**	100%

At these 100%-success settings, key recovery took approximately 4 seconds for HAWK-256 with 11 signatures, 20 seconds for HAWK-512 with 14 signatures, and 157 seconds for HAWK-1024 with 14 signatures. The runtime is dominated by solving the linear program.

5.3.2 Category III: Noisy sign leakage. Category III attacks operate with noisy sign observations in which each sign bit is independently corrupted with probability p. The number of signatures is controlled through a confidence factor C, and larger values of C require proportionally more signatures to compensate for noise and dimensional effects. Table 4 summarizes the results across the tested values of p and C. Each entry reports the number of signatures used followed by the empirical success rate, written in the format "signatures used / success rate".

The results show that increasing the confidence factor C improves the empirical success rate at every tested noise level. At the highest tested noise rate, $p = 0.40$, HAWK-256 achieves 100% success using 2,044 signatures with $C = 1.5$, HAWK-512 achieves 100% success using 6,036 signatures with $C = 2$, and HAWK-1024 achieves 100% success using 13,397 signatures with $C = 2$. Key recovery at these settings took approximately 5 seconds, 20 seconds, and 178 seconds, respectively. Thus, under the tested independent bit-flip model, the recovery algorithm can compensate for substantial sign errors by increasing the number of signatures.

5.4 Parity ablation

The preceding experiments establish that sign leakage alone is sufficient for key recovery, while the measured-leakage results above show that parity can also be extracted reliably from power traces. We therefore ask how much this additional hint can reduce the required number of signatures beyond the sign-only baseline. The purpose of this ablation is to isolate that incremental benefit, rather than to evaluate errors in parity extraction.

Accordingly, we consider an idealized setting in which the adversary receives perfect coefficient parity in addition to exact signs. This subsection reports the resulting reduction, while the underlying parity-assisted reconstruction is detailed in Appendix F: Appendix F.1 embeds the parity hint as one additional column in the

Table 4: Category III success rates over 100 trials under simulated i.i.d. sign-bit flips. Each entry is reported as "signatures used / success rate".

Noise rate p	$C = 0.5$	$C = 1$	$C = 1.5$	$C = 2$
HAWK-256				
0.00	28 / 92%	55 / 100%	82 / 100%	110 / 100%
0.10	43 / 42%	86 / 100%	128 / 100%	172 / 100%
0.20	76 / 6%	152 / 92%	228 / 100%	304 / 100%
0.30	171 / 1%	341 / 79%	511 / 100%	682 / 100%
0.40	682 / 0%	1363 / 79%	2044 / 100%	2726 / 100%
HAWK-512				
0.00	61 / 95%	122 / 100%	183 / 100%	244 / 100%
0.10	95 / 49%	190 / 100%	285 / 100%	380 / 100%
0.20	168 / 10%	336 / 99%	504 / 100%	672 / 100%
0.30	378 / 1%	756 / 85%	1134 / 99%	1512 / 100%
0.40	1509 / 0%	3018 / 81%	4527 / 98%	6036 / 100%
HAWK-1024				
0.00	134 / 97%	268 / 100%	402 / 100%	536 / 100%
0.10	210 / 55%	420 / 100%	628 / 100%	838 / 100%
0.20	374 / 10%	748 / 99%	1117 / 100%	1489 / 100%
0.30	838 / 0%	1676 / 89%	2512 / 100%	3350 / 100%
0.40	3350 / 0%	6700 / 82%	10048 / 99%	13397 / 100%

Category II LP, while Appendix F.2 develops the parity-aware Category III rounding argument and shows that it reduces the sufficient signature bound by a factor of four.

For each signature count $k \in \{3, \ldots, 8\}$, we run 100 trials under the same experimental setting and report the parity-assisted success rate in Table 5. The sign-only results in Table 3 provide the baseline.

Table 5: Parity-assisted Category II success rates over 100 trials.

	Signatures k					
Parameter	3	4	5	6	7	8
HAWK-256	7%	**100%**	100%	100%	100%	100%
HAWK-512	0%	13%	**100%**	100%	100%	100%
HAWK-1024	0%	0%	**100%**	100%	100%	100%

At $k = 3$, the success rates are 7%, 0%, and 0% for HAWK-256, HAWK-512, and HAWK-1024, respectively. At $k = 4$, they become 100%, 13%, and 0%, and all three parameter sets achieve 100% success from $k = 5$ onward. Thus perfect parity lowers the first tested count with observed 100% success from 11 to 4 for HAWK-256 and from 14 to 5 for both HAWK-512 and HAWK-1024.

Noisy parity provides a related recovery route for Category III. When ordinary rounding fails even though $\|\widehat{f} - f\|_\infty < 1$, recovering the required coefficient parities reduces to the structured LPN instance described in Appendix F.2. Its noise vector is exactly the parity-extraction error, so a low parity-error rate gives sparse LPN noise. Solving this auxiliary instance supplies $\overline{f}$ and enables parity-constrained rounding to recover cases missed by ordinary rounding.

5.5 Comparison with Prior HAWK Attacks

Table 6 compares our attacks with the closest prior HAWK attacks based on sampler-output leakage. The algebraic key-exposure attack of Rahmani et al. [18] is not included because it assumes exact exposure of one long-term secret-key polynomial rather than leakage of an ephemeral sampler output. Its signature requirement and robustness to leakage errors are therefore not directly comparable to the attacks considered in the table.

The comparison focuses on the assumed leakage, target signature requirements, noise handling, and key-recovery methods. Runtimes are omitted because the implementations and evaluation platforms differ. Noise handling refers to errors in the leakage supplied to the key-recovery algorithm.

Under full sampler-output leakage, Guerreau and Rossi require two target signatures, whereas Rahmani et al. and our Category I attack require one. Rahmani et al. recover the secret basis sequentially, while our unified identity $M_{\mathbf{w}} = M_{\mathbf{x}}B$ recovers the complete basis and also underlies the constraint and feature constructions of Categories II and III. Unlike the prior sampler-output attacks, Categories II and III recover the key from exact or noisy coefficient signs without requiring complete coefficient values.

6 Discussion

Relation to recent public-key cryptanalysis. Recent public-key cryptanalysis changes the mathematical security context in which our results should be interpreted. It reduces HAWK-n key recovery from public information to polynomially many exact-SVP instances of dimension at most $n/2 + 1$ and requires neither signing executions nor physical leakage [21]. Although the reduction itself is polynomial time, its concrete cost is governed by solving the resulting exact-SVP instances, for which no polynomial-time algorithm is currently known. Our work addresses a distinct implementation security setting in which partial physical leakage from the Gaussian sampler is combined with public signing information and HAWK's Gram-matrix relations. Under the stated leakage models, our recovery algorithms run in polynomial time in the HAWK dimension and the number of observed signatures, while requiring access to physical leakage during signing. Increasing the dimension within the rank-2 structure considered here does not by itself remove the algebraic relations underlying our recovery methods, provided that the assumed leakage remains available. Our algorithms are specific to HAWK's algebraic structure, but more broadly demonstrate how coarse or erroneous leakage can be amplified through public algebraic relations into full key recovery. The measured HAWK-1024 experiment demonstrates this trace-to-key path using physically extracted exact signs for one fixed key, whereas robustness to noisy signs is evaluated separately under the controlled independent sign-error model.

Realism of the independent sign-bit flip model. Category III uses i.i.d. sign-bit flips as an analytically tractable baseline rather than claiming that physical classification errors are independent. In practice, trace misalignment, classifier bias, and device-dependent effects may produce correlated or non-uniform errors outside our theoretical guarantees. Injecting independent Gaussian noise into traces, as in Section 5.1, also does not establish independence of the

Table 6: Comparison of sampler-leakage key-recovery attacks on HAWK-1024.

Attack	Assumed leakage	Signatures	Noise handling	key-recovery method
Prior work				
Guerreau–Rossi [9]	Exact values for all or approximately 98% of $\mathbf{x}$, reconstructed from sign-, parity-, and value-dependent leakage	2[a]	Not modeled	Matrix inversion, preceded by enumeration when $\mathbf{x}$ is incomplete
Rahmani et al. [19]	Exact $\mathbf{x}$ and the corresponding public $\mathbf{w}$ from one signature	1	Not modeled	Linear recovery of one basis column, followed by reconstruction of the remaining column
HAWK with Hint				
Ours, Category I	Exact $\mathbf{x}$ and the corresponding public transcript	1	Not modeled	Unified recovery using $B = M_{\mathbf{x}}^{-1} M_{\mathbf{w}}$
Ours, Category II	Exact per-coefficient signs of $\mathbf{x}$	14	Exact signs	Spectral symmetry alignment and linear programming
Ours, Category III	Per-coefficient signs corrupted by independent Bernoulli-p flips	420 at $p = 0.10$ and 13,397 at $p = 0.40$	$p < 1/2$, tested up to $p = 0.40$	Spectral symmetry alignment, sign aggregation, and rounding

[a] The count refers to target signatures and excludes traces used to construct the profiled template.

resulting classification errors. The $p = 0.10$ experiment is comparable to the 7.7% error rate in Table 1 only at the marginal level, while $p = 0.40$ is a robustness stress test rather than a measured condition. Our measured HAWK-1024 experiment recovered all 28,672 signs correctly, validating the error-free recovery pipeline but providing no error distribution for assessing correlations. Evaluating the attack under correlated errors remains future work.

Interaction with countermeasures. Guerreau and Rossi proposed lightweight ad hoc modifications to reduce sign-, parity-, and value-dependent leakage from HAWK's Gaussian sampler [9]. Their modifications prevented the attacks evaluated on their platform from recovering the key with essentially no performance overhead. However, they provide no formal guarantee and leave residual leakage in the sign application and norm computation. Repeating the isolated-coefficient experiment of Section 5.1 on the modified implementation, we correctly recovered the sign from all 100 measured traces using the same acquisition and analysis procedure. Thus, under our measurement setting, the modifications do not eliminate the sign leakage required by Categories II and III.

Abou Haidar et al. proposed Maskaglia, which provides masked samplers for HAWK's two Gaussian distributions with centers $c \in \{0, 1/2\}$ [1]. They prove Probe-Isolating Non-Interference in the t-probing model, where an adversary observes at most t internal values. This property supports secure composition and hides a sampled coefficient, including its sign, while it remains shared. Maskaglia could address the leakage required by our attacks if masking is maintained through the secret-dependent sample selection and all subsequent operations involving the sampled value. Intermediate unmasking could reintroduce exploitable sign leakage. Their HAWK evaluation is based primarily on masked-gate costs and does not include a complete signing implementation or physical leakage evaluation. Its effectiveness against our attacks in a concrete implementation therefore remains to be experimentally evaluated.

Acknowledgements

This work was supported by Korea Internet & Security Agency (KISA) grant funded by the Korea government (PIPC) (No.2780000036, Masters and Doctoral Program for Advanced Talent Development in Personal Data Protection and Responsible Use) and this work was supported by Korea University grants K2603241.

References

[1] Calvin Abou Haidar, Thomas Espitau, Clément Hoffmann, and Mehdi Tibouchi. 2026. Maskaglia: A New, Efficient Approach to Masked Discrete Gaussian Sampling. *Cryptology ePrint Archive* (2026).

[2] 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. 2024. *HAWK*. Technical Report. National Institute of Standards and Technology. available at https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures.

[3] Dana Dachman-Soled, Léo Ducas, Huijing Gong, and Mélissa Rossi. 2020. LWE with side information: Attacks and concrete security estimation. In *Annual international cryptology conference*. Springer, 329–358.

[4] Léo Ducas, Eamonn W Postlethwaite, Ludo N Pulles, and Wessel van Woerden. 2022. Hawk: Module LIP makes lattice signatures fast, compact and simple. In *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 65–94.

[5] Thomas Espitau, Pierre-Alain Fouque, Benoît Gérard, and Mehdi Tibouchi. 2017. Side-channel attacks on BLISS lattice-based signatures: Exploiting branch tracing against strongswan and electromagnetic emanations in microcontrollers. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 1857–1874.

[6] Pierre-Alain Fouque, Paul Kirchner, Mehdi Tibouchi, Alexandre Wallet, and Yang Yu. 2020. Key recovery from gram–schmidt norm leakage in hash-and-sign signatures over NTRU lattices. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 34–63.

[7] Leon Groot Bruinderink, Andreas Hülsing, Tanja Lange, and Yuval Yarom. 2016. Flush, gauss, and reload–a cache attack on the bliss lattice-based signature scheme. In *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 323–345.

[8] Morgane Guerreau, Ange Martinelli, Thomas Ricosset, and Mélissa Rossi. 2022. The hidden parallelepiped is back again: Power analysis attacks on falcon. *IACR Transactions on Cryptographic Hardware and Embedded Systems* (2022), 141–164.

[9] Morgane Guerreau and Mélissa Rossi. 2024. A Not So Discrete Sampler: Power Analysis Attacks on HAWK signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2024, 4 (2024), 156–178.

[10] Julius Hermelink, Silvan Streit, Erik Mårtensson, and Richard Petri. 2025. A Generic Framework for Side-Channel Attacks Against LWE-Based Cryptosystems. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 3–32.

[11] Emre Karabulut and Aydin Aysu. 2021. Falcon down: Breaking falcon post-quantum signature scheme through side-channel attacks. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 691–696.
[12] Suhri Kim and Seokhie Hong. 2018. Single trace analysis on constant time CDT sampler and its countermeasure. *Applied Sciences* 8, 10 (2018), 1809.
[13] Xiuhan Lin, Shiduo Zhang, Yang Yu, Weijia Wang, Qidi You, Ximing Xu, and Xiaoyun Wang. 2025. Thorough power analysis on falcon gaussian samplers and practical countermeasure. In *IACR International Conference on Public-Key Cryptography*. Springer, 229–258.
[14] Qian Lu, Yansong Feng, and Yanbin Pan. 2025. Solving LWE with Independent Hints about Secret and Errors. *Cryptology ePrint Archive* (2025).
[15] Alexander May and Julian Nowakowski. 2023. Too many hints–when lll breaks lwe. In *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 106–137.
[16] Chris Peikert. 2010. An efficient and parallel Gaussian sampler for lattices. In *Annual Cryptology Conference*. Springer, 80–97.
[17] Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. 2022. *FALCON*. Technical Report. National Institute of Standards and Technology. available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.
[18] Nour-eddine Rahmani, Taoufik Serraj, and Moulay Chrif Ismaili. 2025. An algebraic key-exposure attack on the digital signature HAWK. *Gulf Journal of Mathematics* 19, 2 (2025), 384–397.
[19] Nour-eddine Rahmani, Taoufik Serraj, and Moulay Chrif Ismaili. 2025. An Improvement of the Congruence Solver of Lattice Isomorphism Problem over Totally Real Number Fields and Applications. In *International Conference on Cryptology in Africa*. Springer, 147–163.
[20] Honglin Shao, Yuejun Liu, Mingyao Shao, and Yongbin Zhou. 2025. NTRU with Hints: Recovering NTRU Secret Keys from Partial Leakage. *Cryptology ePrint Archive* (2025).
[21] Zygimantas Straznickas and Stephen A Weis. 2026. HAWK-n Key Recovery Reduces to SVP in Dimension $n/2 + 1$. *Cryptology ePrint Archive* (2026).
[22] James G Wendel. 1962. A problem in geometric probability. *Math. Scand.* 11, 1 (1962), 109–111.
[23] Han Wu, Xiaoyun Wang, and Guangwu Xu. 2022. On the Dual Attack of LWE Schemes in the Presence of Hints. *Cryptology ePrint Archive* (2022).
[24] Shiduo Zhang, Xiuhan Lin, Yang Yu, and Weijia Wang. 2023. Improved power analysis attacks on falcon. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 565–595.

Generative AI Usage

This paper was edited for grammar using Claude Code. Claude Code was also used to assist with the code included in the public artifact, such as code organization, readability improvements, and implementation-level editing.

A Open Science

An artifact containing the implementation and scripts needed to reproduce the experiments is available at: https://github.com/ksprsk/hawkside. The artifact includes the attack implementations, experiment scripts, and documentation for reproducing the results reported for HAWK.

B Ethical Considerations

Our work focuses on side-channel analysis of post-quantum signature implementations on microcontroller boards that are fully owned and controlled by the authors. No human subjects or personal data were involved. We did not attack any deployed production systems and all experiments were conducted on lab setups and reference or research implementations. The findings aim to improve the robustness of future implementations by highlighting potential side-channel vulnerabilities.

C Proofs for Section 4

In this appendix, we provide detailed proofs for the lemmas and theorems stated in Section 4. These results formalize the algebraic and probabilistic relations used in the main text and provide the theoretical foundation for the key-recovery algorithms developed in Sections 4.2–4.5.

Lemma 4.1. For $B, Q, \mathbf{x}$, and $\widehat{\mathbf{w}}$ defined in Section 4.1, the matrices $M_{\mathbf{x}}$ and $M_{\widehat{\mathbf{w}}}$ satisfy

$$M_{\widehat{\mathbf{w}}} B^{-1} = M_{\mathbf{x}}, \qquad M_{\widehat{\mathbf{w}}} = M_{\mathbf{x}} B.$$

Proof. Recall that $Q = B^\star B$, $\mathbf{x} = B\widehat{\mathbf{w}}$, and

$$M_{\mathbf{x}} = \begin{bmatrix} x_0^\star & x_1^\star \\ -x_1 & x_0 \end{bmatrix}, \qquad M_{\widehat{\mathbf{w}}} = \begin{bmatrix} (\widehat{\mathbf{w}}^\star Q)_0 & (\widehat{\mathbf{w}}^\star Q)_1 \\ -\widehat{w}_1 & \widehat{w}_0 \end{bmatrix}.$$

Using $Q = B^\star B$ and $\mathbf{x} = B\widehat{\mathbf{w}}$, we obtain

$$\widehat{\mathbf{w}}^\star Q = \widehat{\mathbf{w}}^\star B^\star B = (B\widehat{\mathbf{w}})^\star B = \mathbf{x}^\star B.$$

Thus the first row of $M_{\widehat{\mathbf{w}}}$ is $\mathbf{x}^\star B$, and

$$\big((\widehat{\mathbf{w}}^\star Q)_0, (\widehat{\mathbf{w}}^\star Q)_1\big) B^{-1} = \mathbf{x}^\star B B^{-1} = (x_0^\star, x_1^\star),$$

which is the first row of $M_{\mathbf{x}}$.

For the second row, since $\det(B) = 1$ and $\mathbf{x} = B\widehat{\mathbf{w}}$, a direct calculation gives

$$(-\widehat{w}_1, \widehat{w}_0) B^{-1} = (-x_1, x_0),$$

which is the second row of $M_{\mathbf{x}}$. Therefore,

$$M_{\widehat{\mathbf{w}}} B^{-1} = M_{\mathbf{x}}.$$

Multiplying on the right by B yields

$$M_{\widehat{\mathbf{w}}} = M_{\mathbf{x}} B,$$

as claimed. □

Lemma 4.2. For $\mathbf{w} = \varkappa\, \widehat{\mathbf{w}}$ with $\varkappa \in \{\pm 1\}$, the corresponding M-matrices satisfy

$$M_{\mathbf{w}} = \varkappa\, M_{\widehat{\mathbf{w}}}.$$

Proof. By definition of the symmetry-breaking step,

$$\mathbf{w} = \varkappa\, \widehat{\mathbf{w}}, \qquad \varkappa \in \{\pm 1\},$$

and therefore

$$\mathbf{w}^\star = \varkappa\, \widehat{\mathbf{w}}^\star, \qquad \mathbf{w}^\star Q = \varkappa\, \widehat{\mathbf{w}}^\star Q.$$

Hence the first row of $M_{\mathbf{w}}$ is $\varkappa$ times the first row of $M_{\widehat{\mathbf{w}}}$:

$$\big((\mathbf{w}^\star Q)_0, (\mathbf{w}^\star Q)_1\big) = \varkappa\big((\widehat{\mathbf{w}}^\star Q)_0, (\widehat{\mathbf{w}}^\star Q)_1\big).$$

For the second row,

$$(-w_1, w_0) = (-\varkappa \widehat{w}_1, \varkappa \widehat{w}_0) = \varkappa(-\widehat{w}_1, \widehat{w}_0).$$

Thus every row of $M_{\mathbf{w}}$ is multiplied by $\varkappa$, so

$$M_{\mathbf{w}} = \varkappa\, M_{\widehat{\mathbf{w}}},$$

which proves the lemma. □

Lemma 4.3. The determinants of the matrices $M_{\mathbf{x}}$, $M_{\widehat{\mathbf{w}}}$, and $M_{\mathbf{w}}$ are equal and nonzero:

$$\det(M_{\mathbf{w}}) = \det(M_{\widehat{\mathbf{w}}}) = \det(M_{\mathbf{x}}) = \mathbf{x}^\star\mathbf{x} \neq 0.$$

Proof. By the 2×2 determinant formula,

$$\det(M_{\mathbf{x}}) = x_0^\star x_0 + x_1^\star x_1 = \mathbf{x}^\star\mathbf{x}.$$

Since $\mathbf{x} \neq \mathbf{0}$ in a valid signature, this quantity is nonzero.

From Lemma 4.1,

$$M_{\widehat{\mathbf{w}}} B^{-1} = M_{\mathbf{x}},$$

hence

$$\det(M_{\widehat{\mathbf{w}}}) \det(B^{-1}) = \det(M_{\mathbf{x}}).$$

Because $\det(B) = 1$, we also have $\det(B^{-1}) = 1$, so

$$\det(M_{\widehat{\mathbf{w}}}) = \det(M_{\mathbf{x}}).$$

From Lemma 4.2,

$$M_{\mathbf{w}} = \varkappa M_{\widehat{\mathbf{w}}}, \qquad \varkappa \in \{\pm 1\}.$$

Therefore,

$$\det(M_{\mathbf{w}}) = \det(\varkappa M_{\widehat{\mathbf{w}}}) = \varkappa^2 \det(M_{\widehat{\mathbf{w}}}) = \det(M_{\widehat{\mathbf{w}}}).$$

Combining the identities proves the claim. □

Theorem 4.4. Let $B \in \mathrm{GL}_2(R_n)$ be the secret key of HAWK and let $Q = B^\star B$ be the public key. Suppose that for one signing execution the adversary obtains both $\mathbf{x} \in R_n^2$ and the canonical vector $\mathbf{w} \in R_n^2$, where

$$\mathbf{w} = \varkappa\, \widehat{\mathbf{w}} \quad \text{for some unknown } \varkappa \in \{\pm 1\}, \qquad \mathbf{x} = B\widehat{\mathbf{w}}.$$

Then the adversary can recover either B or $-B$ in polynomial time.

Proof. By Lemmas 4.2 and 4.1,

$$M_{\mathbf{w}} = \varkappa\, M_{\mathbf{x}} B.$$

By Lemma 4.3, the matrix $M_{\mathbf{x}}$ is invertible. Hence the adversary can compute

$$\widetilde{B} := M_{\mathbf{x}}^{-1} M_{\mathbf{w}} = \varkappa\, B \in \{B, -B\}.$$

This requires only matrix inversion and multiplication over K_n, and therefore runs in polynomial time. □

Lemma 4.6. Let $\boldsymbol{\ell}^{(j)}$ and $\mathbf{z}^{(j)}$ be as in Definition 4.5. Then:

(1) **Algebraic relation.** Using Lemmas 4.1 and 4.2, we have

$$\boldsymbol{\ell}^{(j)} = \varkappa^{(j)}\, B^\star \mathbf{z}^{(j)}.$$

(2) **Statistical moments.** The canonical embedding components are approximately distributed as

$$\sigma(\alpha^{(j)})_k \sim \mathcal{CN}\Big(2n\mu,\ 8n^2\sigma_{\mathrm{sign}}^2\Big), \qquad \sigma(\beta^{(j)})_k \sim \mathcal{CN}\Big(0,\ 8n^2\sigma_{\mathrm{sign}}^2\Big),$$

for all $k \in \{0, \ldots, n-1\}$, where

$$\mu = (1-2p)\mu_{\mathrm{sign}}.$$

Proof. For the algebraic relation, take Hermitian transposes in the identities of Lemmas 4.1 and 4.2, and multiply by

$$\begin{pmatrix} (I_0^{(j)})^\star \\ -I_1^{(j)} \end{pmatrix}.$$

This gives

$$\boldsymbol{\ell}^{(j)} = \varkappa^{(j)} B^\star \mathbf{z}^{(j)}.$$

For the statistical moments, independence implies that only diagonal terms contribute to the expectation in the canonical embedding. Under the noisy-sign model,

$$\mathbb{E}[I_0^{(j)}[\ell] x_0^{(j)}[\ell]] = \mu,$$

so

$$\mathbb{E}[\sigma(\alpha^{(j)})_k] = 2n\mu.$$

A variance decomposition into diagonal and off-diagonal terms yields

$$\mathrm{Var}\big(\sigma(\alpha^{(j)})_k\big) = 8n^2\sigma_{\mathrm{sign}}^2 - 2n\mu^2 \approx 8n^2\sigma_{\mathrm{sign}}^2,$$

where the lower-order term is negligible for large n. Similarly,

$$\mathbb{E}[\sigma(\beta^{(j)})_k] = 0, \qquad \mathrm{Var}\big(\sigma(\beta^{(j)})_k\big) = 8n^2\sigma_{\mathrm{sign}}^2.$$

By a central-limit approximation, both quantities are approximately complex Gaussian with the stated parameters. □

Auxiliary concentration lemma. Let $\psi \in \mathbb{R}^k$ be a random vector with $\|\psi\|_2 = 1$ whose entries behave like normalized independent Gaussian coordinates. Then with probability $1 - O(k^{-c})$ for any constant $c > 0$,

$$\|\psi\|_\infty = \Theta\left(\sqrt{\frac{\log k}{k}}\right).$$

Proof. For a normalized random vector with approximately i.i.d. Gaussian coordinates, each coordinate has variance of order $1/k$. The maximum of k such coordinates is therefore of order

$$\sqrt{\frac{2\log k}{k}}.$$

Standard concentration bounds then imply

$$\|\psi\|_\infty = \Theta\left(\sqrt{\frac{\log k}{k}}\right)$$

with probability $1 - O(k^{-c})$. □

Lemma 4.7. Let $G_0 \in \mathbb{R}^{k\times k}$ have entries

$$(G_0)_{ij} := \mathrm{T}\big((\boldsymbol{\ell}^{(i)})^\star Q^{-1}\boldsymbol{\ell}^{(j)}\big),$$

where $\boldsymbol{\ell}^{(j)}$ and $\mathbf{z}^{(j)}$ are as in Definition 4.5. Then

$$(G_0)_{ij} = \varkappa^{(i)}\varkappa^{(j)}\,\mathrm{T}\big((\mathbf{z}^{(i)})^\star \mathbf{z}^{(j)}\big).$$

In particular, under the noisy-sign model of Lemma 4.6 with flip rate $p < \frac{1}{2}$,

$$\mathbb{E}[G_0] = \sigma_0 I_k + \lambda\,\psi\psi^\mathsf{T}, \qquad \psi = (\varkappa^{(1)},\dots,\varkappa^{(k)})^\mathsf{T},$$

with

$$\lambda = 4n^2\mu^2, \qquad \sigma_0 = 16n^2\sigma_{\mathrm{sign}}^2,$$

where $\mu = (1-2p)\mu_{\mathrm{sign}}$.

Proof. From the first part of Lemma 4.6 and the identity

$$BQ^{-1}B^\star = I_2,$$

we obtain

$$(G_0)_{ij} = \varkappa^{(i)}\varkappa^{(j)}\,\mathrm{T}\big((\mathbf{z}^{(i)})^\star \mathbf{z}^{(j)}\big).$$

For $i \neq j$, independence implies

$$\mathbb{E}\big[\mathrm{T}((\alpha^{(i)})^\star\alpha^{(j)})\big] = 4n^2\mu^2, \qquad \mathbb{E}\big[\mathrm{T}((\beta^{(i)})^\star\beta^{(j)})\big] = 0,$$

so

$$\mathbb{E}[(G_0)_{ij}] = \varkappa^{(i)}\varkappa^{(j)}\cdot 4n^2\mu^2.$$

For $i = j$, using second moments gives

$$\mathbb{E}[(G_0)_{ii}] = 4n^2\mu^2 + 16n^2\sigma_{\mathrm{sign}}^2.$$

Therefore,

$$\mathbb{E}[G_0] = \sigma_0 I_k + \lambda\,\psi\psi^\mathsf{T},$$

with the stated values of λ and σ_0. □

Theorem 4.8. Given k pairs $\{(\mathbf{w}^{(j)}, I^{(j)})\}_{j=1}^k$ whose signs follow the noisy-sign model, if

$$k \geq \frac{8\sigma_{\mathrm{sign}}^2(\log k)^2}{n\mu^2}\left(1 + \frac{2\sigma_{\mathrm{sign}}^2}{\mu^2}\right),$$

where $\mu = (1-2p)\mu_{\mathrm{sign}}$, then the algorithm runs in polynomial time

$$T_1 = O(Tkn),$$

and for each $j = 1,\dots,k$, with probability $1 - O(1/k)$, the output of Algorithm 1 satisfies

$$\mathrm{sign}(\widehat{\psi}_j) = \varkappa^{(j)}.$$

Proof. By Lemma 4.7,

$$\mathbb{E}[G_0] = \sigma_0 I_k + \lambda\psi\psi^\mathsf{T},$$

whose eigengap is

$$\Delta = k\lambda.$$

Let

$$E = G_0 - \mathbb{E}[G_0].$$

If $\widehat{\psi}$ denotes the top unit eigenvector of G_0, then the Davis–Kahan theorem gives

$$\left\|\widehat{\psi} - \frac{\psi}{\sqrt{k}}\right\|_2 \leq \frac{\|E\|_{\mathrm{op}}}{k\lambda}.$$

Using matrix Bernstein, one obtains

$$\|E\|_{\mathrm{op}} \lesssim \sqrt{k\sigma^2\log k},$$

where

$$\sigma^2 = 64n^3\sigma_{\mathrm{sign}}^2\left(\mu^2 + 2\sigma_{\mathrm{sign}}^2\right).$$

To recover all signs correctly, it suffices that

$$\left\|\widehat{\psi} - \frac{\psi}{\sqrt{k}}\right\|_\infty < \frac{1}{\sqrt{k}}.$$

By the auxiliary ℓ_∞ concentration lemma,

$$\left\|\widehat{\psi} - \frac{\psi}{\sqrt{k}}\right\|_\infty \lesssim \left\|\widehat{\psi} - \frac{\psi}{\sqrt{k}}\right\|_2 \sqrt{\frac{\log k}{k}}.$$

Combining these inequalities yields the stated lower bound on k.

For the complexity, the Lanczos method computes the leading eigenvector using matrix-vector products with

$$G_0 = A\,\mathrm{Lift}(Q^{-1})A^\top,$$

without explicitly forming G_0. Each iteration costs $O(kn)$, and thus the total runtime is $O(Tkn)$. □

Theorem 4.11. Let B be the secret key and $Q = B^\star B$ the public key of HAWK. Given k pairs $\{(\widehat{\mathbf{w}}^{(j)}, I^{(j)})\}_{j=1}^k$ with exact sign observations ($p = 0$), the success probability of recovering B in time $T_2 = O(n^3k)$ is determined by whether the observed hyperplanes are in general position. Specifically,

$$\Pr[\text{key recovery} \mid k] = \mathbb{E}_{m\sim\mathrm{Bin}(2nk,p_0)}[P_{2n-1}(m)],$$

where

$$p_0 = \tfrac{1}{2}\Pr_{Z\sim D_{\mathbb{Z},\sigma_{\mathrm{sign}}}}[Z = 0]$$

and

$$P_d(m) := 1 - 2^{-m+1}\sum_{j=0}^{d-1}\binom{m-1}{j}$$

is the probability that m random hyperplanes through the origin in $\mathbb{R}^d$ are in general position [22].

Proof. Let

$$\mathbf{z}_{\mathrm{key}} := \mathrm{Vec}(G, -g) \in \mathbb{Z}^{2n}$$

be the first column of B^{-1}. From the observed signs, construct constraint matrices

$$A^{(j)} = \mathrm{Diag}(\mathbf{v}^{(j)})\cdot\mathrm{Lift}(M_{\widehat{\mathbf{w}}}^{(j)}).$$

Stacking them gives

$$A \in \mathbb{R}^{2nk\times 2n},$$

with feasible cone

$$\mathcal{K} = \{\mathbf{z}\in\mathbb{R}^{2n} : A\mathbf{z} \succeq \mathbf{0}\}.$$

Let m denote the number of zero coefficients. The feasible cone collapses to a ray if and only if the corresponding zero-coefficient hyperplanes are in general position. By Wendel's theorem, this occurs with probability

$$P_{2n-1}(m).$$

Since

$$m \sim \mathrm{Bin}(2nk, p_0),$$

the overall success probability is

$$\Pr[\text{key recovery} \mid k] = \mathbb{E}_{m\sim\mathrm{Bin}(2nk,p_0)}[P_{2n-1}(m)].$$

Constructing A costs $O(n^2k)$, and solving the resulting LP by the interior-point method costs $O(n^3k)$, yielding the stated complexity. □

Lemma 4.13. Let $\boldsymbol{\ell}^{(j)}$ and $\mathbf{z}^{(j)}$ be the observable and theoretical feature vectors. The aligned aggregates

$$\boldsymbol{\ell}_\Sigma = \sum_{j=1}^{k} \varkappa^{(j)} \boldsymbol{\ell}^{(j)}, \qquad \mathbf{z}_\Sigma = \sum_{j=1}^{k} \mathbf{z}^{(j)}$$

satisfy the algebraic relation

$$\boldsymbol{\ell}_\Sigma = B^\star \mathbf{z}_\Sigma.$$

Proof. This follows immediately by summing the identities

$$\boldsymbol{\ell}^{(j)} = \varkappa^{(j)} B^\star \mathbf{z}^{(j)}$$

from Lemma 4.6. □

Theorem 4.14. Suppose the symmetry-breaking signs $\{\varkappa^{(j)}\}_{j=1}^{k}$ have been correctly recovered. If

$$k \geq \frac{12C\eta\sigma_{\text{sign}}^2 \log n}{\mu^2},$$

then the aggregation and slot-wise normalization procedure recovers $f^\star$ and hence the secret basis B in time

$$T_3 = O(kn + n \log n)$$

via coefficient-wise rounding, with probability at least

$$1 - n^{1-\Omega(C)},$$

where $C > 0$ is an absolute constant.

Proof. The algorithm is given in Algorithm 3. By Lemma 4.13,

$$\boldsymbol{\ell}_\Sigma = B^\star \mathbf{z}_\Sigma,$$

where

$$\mathbf{z}_\Sigma = \begin{pmatrix} \alpha_\Sigma \\ \beta_\Sigma \end{pmatrix}.$$

In the high-SNR regime,

$$\alpha_\Sigma \approx 2kn\mu + Z_\alpha, \qquad \beta_\Sigma \approx Z_\beta,$$

with

$$\mathbb{E}[|Z_\alpha|^2] = \mathbb{E}[|Z_\beta|^2] = 8kn^2\sigma_{\text{sign}}^2.$$

After normalization, the first component concentrates near 1 and the second component near 0, so the normalized aggregate approximates the first column of $B^\star$.

A variance calculation shows that the resulting reconstruction error $\widehat{\varepsilon}$ after inverse embedding satisfies

$$\mathbb{E}\left[\|\widehat{\varepsilon}\|_2^2\right] \approx \frac{3n\eta\,\sigma_{\text{sign}}^2}{2k\mu^2}.$$

Using concentration from ℓ_2 to ℓ_∞, one gets

$$\|\widehat{\varepsilon}\|_\infty \leq \sqrt{\frac{3C\eta\,\sigma_{\text{sign}}^2 \log n}{k\mu^2}}.$$

Coefficient-wise rounding succeeds whenever this quantity is smaller than $1/2$, which yields the stated lower bound on k.

The runtime consists of $O(kn)$ aggregation and $O(n \log n)$ FFT-based operations, giving total complexity $O(kn + n \log n)$. □

D Interior-Point Method

This appendix describes the primal-dual interior-point method used to solve the linear program in Category II from Section 4.4.

D.1 Linear Programming Formulation

Using the constraint matrix

$$A \in \mathbb{R}^{m \times 2n}, \qquad m = 2nk,$$

from Definition 4.9, we seek

$$\mathbf{z} \in \mathbb{R}^{2n}$$

satisfying

$$A\mathbf{z} \succeq \mathbf{0}.$$

Since these constraints define a cone through the origin, we add the normalization constraint

$$\langle \mathbf{c}_{\text{sum}}, \mathbf{z} \rangle = 1, \qquad \mathbf{c}_{\text{sum}} := \sum_{i=1}^{m} \mathbf{a}_i.$$

The resulting LP is

$$\max_{\mathbf{z}} \langle \mathbf{c}, \mathbf{z} \rangle \quad \text{subject to} \quad A\mathbf{z} \succeq \mathbf{0}, \qquad \langle \mathbf{c}_{\text{sum}}, \mathbf{z} \rangle = 1, \tag{7}$$

where

$$\mathbf{c} \in \mathbb{R}^{2n}$$

is chosen orthogonal to $\mathbf{c}_{\text{sum}}$.

D.2 KKT Conditions and Barrier Method

Introduce slack variables

$$\mathbf{s} \in \mathbb{R}^m,$$

dual variables

$$\mathbf{z}_d \in \mathbb{R}^m,$$

and a Lagrange multiplier

$$\nu \in \mathbb{R}.$$

The KKT conditions are

$$A\mathbf{z} - \mathbf{s} = \mathbf{0}, \qquad -A^\top \mathbf{z}_d + \mathbf{c}_{\text{sum}}\nu = \mathbf{c}, \qquad \langle \mathbf{c}_{\text{sum}}, \mathbf{z} \rangle = 1,$$
$$s_i(\mathbf{z}_d)_i = 0 \quad \forall i, \qquad \mathbf{s}, \mathbf{z}_d \succ \mathbf{0}. \tag{8}$$

The barrier method replaces the exact complementarity condition by

$$s_i(\mathbf{z}_d)_i \approx \sigma\mu,$$

where

$$\mu := \frac{\langle \mathbf{s}, \mathbf{z}_d \rangle}{m}$$

is the duality gap and $\sigma \in (0, 1)$ is a centering parameter.

D.3 Newton System

Applying Newton's method and eliminating $\Delta\mathbf{s}$ and $\Delta\mathbf{z}_d$ yields the reduced system

$$\begin{bmatrix} A^\top W A & \mathbf{c}_{\text{sum}} \\ \mathbf{c}_{\text{sum}}^\top & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{z} \\ \Delta\nu \end{bmatrix} = \begin{bmatrix} -\mathbf{b} \\ -r_e \end{bmatrix}, \tag{9}$$

where

$$W := Z_d S^{-1}$$

is diagonal, with

$$S = \text{diag}(\mathbf{s}), \qquad Z_d = \text{diag}(\mathbf{z}_d).$$

After solving (9), one recovers $\Delta\mathbf{s}$ and $\Delta\mathbf{z}_d$.

Algorithm 4: Primal-Dual Interior-Point Method for the Category II LP

Input: Constraint matrix $A \in \mathbb{R}^{m\times 2n}$
Output: Feasible $\mathbf{z}^* \in \mathbb{R}^{2n}$ with $A\mathbf{z}^* \succeq \mathbf{0}$ and $\langle \mathbf{c}_{\text{sum}}, \mathbf{z}^* \rangle = 1$

Compute $\mathbf{c}_{\text{sum}} \leftarrow \sum_{i=1}^{m} \mathbf{a}_i$
Choose $\mathbf{c} \in \mathbb{R}^{2n}$ orthogonal to $\mathbf{c}_{\text{sum}}$
Initialize $\mathbf{z}$, $\mathbf{s}$, $\mathbf{z}_d$, and ν with $\mathbf{s}, \mathbf{z}_d \succ \mathbf{0}$
Set $\sigma = 0.4$, $\tau = 0.99$, and N_{iter} as the iteration budget
for iter = 1 **to** N_{iter} **do**
  Compute residuals $\mathbf{r}_p$, $\mathbf{r}_d$, and r_e, and the duality gap $\mu = \langle \mathbf{s}, \mathbf{z}_d \rangle / m$
  if $\min(A\mathbf{z}) \geq -\varepsilon(1 + \|\mathbf{z}\|)$ **then**
    return $\mathbf{z}$
  Compute $W = Z_d S^{-1}$, $\mathbf{r}_\mu = SZ_d - \sigma\mu\mathbf{1}$, and $\mathbf{b} = \mathbf{r}_d + A^\top(S^{-1}\mathbf{r}_\mu + W\mathbf{r}_p)$
  Solve (9) for $(\Delta\mathbf{z}, \Delta\nu)$
  Recover $\Delta\mathbf{s} \leftarrow A\Delta\mathbf{z} + \mathbf{r}_p$
  Recover $\Delta\mathbf{z}_d$ by component-wise division from $-\mathbf{r}_\mu - Z_d\Delta\mathbf{s}$ and $\mathbf{s}$
  Compute step sizes α_p and α_d by the fraction-to-boundary rule
  Update $\mathbf{z} \leftarrow \mathbf{z} + \alpha_p\Delta\mathbf{z}$, $\mathbf{s} \leftarrow \mathbf{s} + \alpha_p\Delta\mathbf{s}$, $\mathbf{z}_d \leftarrow \mathbf{z}_d + \alpha_d\Delta\mathbf{z}_d$, and $\nu \leftarrow \nu + \alpha_p\Delta\nu$
return $\mathbf{z}$

D.4 Algorithm

Algorithm 4 summarizes the resulting primal-dual interior-point method. Exploiting the diagonal structure of W reduces the per-iteration complexity from $O(n^3k^3)$ to $O(n^3k)$.

E Empirical Visualization of HAWK's DGS Leakage under Noise

Figure 2 visualizes how the three types of side-channel leakage in HAWK's DGS degrade under increasing noise. The figure is organized into three horizontal subfigures: Figure 2a shows sign leakage, Figure 2b shows parity leakage, and Figure 2c shows increment leakage. Within each row, the four panels correspond to the original trace ($\alpha = 0$) and three noisy cases ($\alpha = 3, 7, 10$), respectively. Each panel is generated from 30 traces collected under the corresponding noise setting.

The horizontal axis denotes the sample index and can be interpreted as a time axis, while the vertical axis represents the measured power consumption. The color of each trace encodes the relevant coefficient information. For example, in the sign-leakage row, negative coefficients are shown in red and positive coefficients in blue, making it possible to visually compare how the corresponding data property influences the observed power traces.

In Figure 2a, the difference between negative and positive coefficients becomes visible around sample point 7729 and remains distinguishable even under relatively high noise. In Figure 2b, the gap between even and odd coefficients emerges around sample point 3097 and also remains visually robust as noise increases. By contrast, Figure 2c shows that increment leakage, observed around sample point 1993, degrades much more rapidly than sign or parity leakage. This layout therefore enables direct visual comparison of how the information content at the three leakage points diminishes under increasing noise.

F Perfect Parity Leakage

The observation $I^{(j)}$ in Definition 3.1 may contain multiple sampler-derived hints. The three categories in the main analysis retain only sign information to establish sign-only sufficiency. Section 5.4 evaluates the additional contribution of perfect coefficient parity. Here we give the corresponding Category II construction and record its exact consequence for the rounding step of Category III.

F.1 Parity-Assisted Category II

Let $\mathbf{a}_\ell$ be row ℓ of the aggregate constraint matrix A from Definition 4.9, and let x_ℓ denote its corresponding sampled coefficient. Define the perfect parity observation

$$\pi_\ell := x_\ell \bmod 2 \in \{0, 1\}, \qquad \ell \in \{1, \ldots, m\}.$$

We stack these observations as

$$\boldsymbol{\pi} := (\pi_1, \ldots, \pi_m)^\top \in \{0, 1\}^m, \qquad m = 2nk.$$

By Property 4.10,

$$(A\mathbf{z}_{\text{key}})_\ell = \langle \mathbf{a}_\ell, \mathbf{z}_{\text{key}} \rangle = |x_\ell|.$$

Perfect parity therefore gives the exact algebraic relation

$$A\mathbf{z}_{\text{key}} = \boldsymbol{\pi} + 2\mathbf{u}, \qquad \mathbf{u} \in \mathbb{Z}^m_{\geq 0}. \tag{10}$$

Equivalently,

$$(A \bmod 2)(\mathbf{z}_{\text{key}} \bmod 2) = \boldsymbol{\pi} \quad \text{over } \mathbb{F}_2.$$

For the Category II LP, we use the nonnegative consequence $A\mathbf{z}_{\text{key}} - \boldsymbol{\pi} \succeq \mathbf{0}$ by embedding the parity offset as one additional column. Define

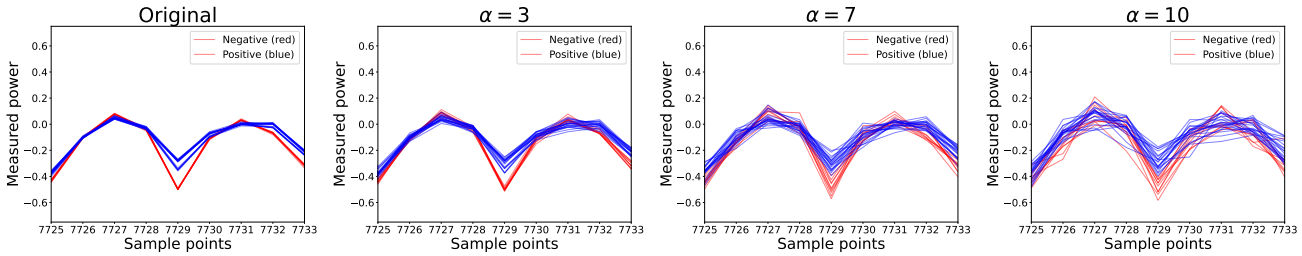
$$A_{\boldsymbol{\pi}} := \begin{bmatrix} A & -\boldsymbol{\pi} \end{bmatrix} \in \mathbb{R}^{m\times(2n+1)}, \qquad \widetilde{\mathbf{z}}_{\text{key}} := \begin{bmatrix} \mathbf{z}_{\text{key}} \\ 1 \end{bmatrix}.$$

Equation (10) then gives

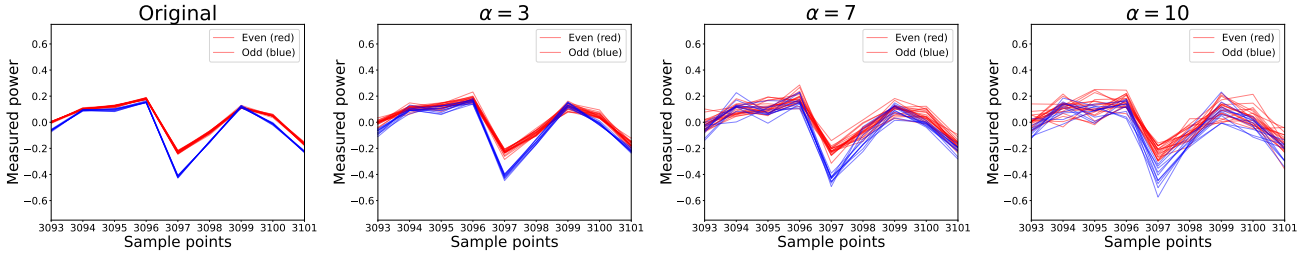
$$A_{\boldsymbol{\pi}}\widetilde{\mathbf{z}}_{\text{key}} = A\mathbf{z}_{\text{key}} - \boldsymbol{\pi} = 2\mathbf{u} \succeq \mathbf{0}.$$

Thus the parity-assisted constraints have the same homogeneous form as the sign-only Category II constraints, with only one extra coordinate. We apply the same LP and interior-point method from Appendix D with $A_{\boldsymbol{\pi}}$ in place of A. The existing post-solution scale-recovery and rounding step of Algorithm 2 is applied to the augmented solution. Its final coordinate is then discarded, and the reconstruction of B proceeds unchanged.

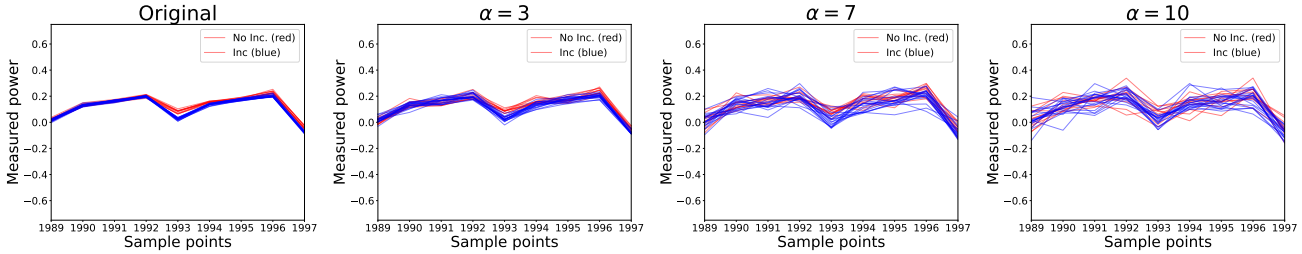
For sign-only recovery, a row is tight when the corresponding coefficient is zero. Under the augmented parity constraints, row ℓ evaluates to $|x_\ell| - \pi_\ell$: an even row is tight at $x_\ell = 0$, while an odd row is tight at $|x_\ell| = 1$. Thus the coefficients inducing tight rows expand from $x_\ell = 0$ to $x_\ell \in \{-1, 0, 1\}$. These additional tight constraints underlie the empirical ablation reported in Section 5.4.



(a) Sign leakage in HAWK's DGS.



(b) Parity leakage in HAWK's DGS.



(c) Increment leakage in HAWK's DGS.

Figure 2: Visualization of sign, parity, and increment leakages in HAWK's DGS under the original trace ($\alpha = 0$) and three noisy levels ($\alpha = 3, 7, 10$). Each row corresponds to one leakage type, and each row contains four panels arranged horizontally according to the noise level.

F.2 Effect on Category III Rounding

Let

$$\overline{R}_n := R_n/2R_n \simeq \mathbb{F}_2[X]/(X^n+1),$$

and let a bar denote reduction modulo 2. For signing execution j,

$$\mathbf{x}^{(j)} = B\widehat{\mathbf{w}}^{(j)} = B\big(h^{(j)} + 2s^{(j)}\big),$$

so perfect coefficient parity gives

$$\overline{\mathbf{x}}^{(j)} = \overline{B}\,\overline{h}^{(j)} \quad \text{over } \overline{R}_n. \tag{11}$$

Writing

$$B = \begin{pmatrix} f & F \\ g & G \end{pmatrix}, \qquad h^{(j)} = (h_0^{(j)}, h_1^{(j)})^{\mathsf{T}},$$

Equation (11) is equivalent to

$$\overline{x}_0^{(j)} = \overline{f}\,\overline{h}_0^{(j)} + \overline{F}\,\overline{h}_1^{(j)},$$
$$\overline{x}_1^{(j)} = \overline{g}\,\overline{h}_0^{(j)} + \overline{G}\,\overline{h}_1^{(j)}.$$

Define

$$H^{(j)} := \begin{bmatrix} \mathrm{Rot}(h_0^{(j)}) & \mathrm{Rot}(h_1^{(j)}) \end{bmatrix} \bmod 2 \in \mathbb{F}_2^{n\times 2n},$$

and

$$\mathbf{b}_0 := \begin{pmatrix} \mathrm{vec}(\overline{f}) \\ \mathrm{vec}(\overline{F}) \end{pmatrix}, \qquad \mathbf{b}_1 := \begin{pmatrix} \mathrm{vec}(\overline{g}) \\ \mathrm{vec}(\overline{G}) \end{pmatrix}.$$

The multiplication convention from Section 2.1 gives

$$H^{(j)}\mathbf{b}_0 = \mathrm{vec}(\overline{x}_0^{(j)}), \qquad H^{(j)}\mathbf{b}_1 = \mathrm{vec}(\overline{x}_1^{(j)}).$$

For k signatures, let

$$H_k := \begin{bmatrix} H^{(1)} \\ \vdots \\ H^{(k)} \end{bmatrix} \in \mathbb{F}_2^{kn\times 2n},$$

and stack the corresponding right-hand sides as $\mathbf{y}_{0,k}, \mathbf{y}_{1,k} \in \mathbb{F}_2^{kn}$. We obtain the two known systems

$$H_k\mathbf{b}_0 = \mathbf{y}_{0,k}, \qquad H_k\mathbf{b}_1 = \mathbf{y}_{1,k}.$$

Since the public $h^{(j)}$ are pseudorandom, H_k is expected to attain full column rank $2n$, a condition directly checkable from the transcript. Gaussian elimination then recovers B mod 2, including $\overline{f}$. No symmetry-breaking alignment is needed because $-1 = 1$ modulo 2.

Let $\widehat{f}$ be the pre-rounding estimate of f in the Category III recovery algorithm (Algorithm 3), and define $\widehat{\varepsilon} := \widehat{f} - f$. Once $\overline{f}[i] \in \{0, 1\}$ is known, ordinary coefficient-wise rounding can be replaced by

$$\mathrm{round}_{\overline{f}[i]}(y) := \overline{f}[i] + 2\,\mathrm{round}\left(\frac{y - \overline{f}[i]}{2}\right)$$

which returns the nearest integer having the recovered parity. It recovers $f[i]$ whenever

$$|\widehat{\varepsilon}[i]| = |\widehat{f}[i] - f[i]| < 1,$$

whereas ordinary integer rounding requires an error smaller than $1/2$.

The proof of Theorem 4.14 gives

$$\|\widehat{\varepsilon}\|_\infty \le \sqrt{\frac{3C\eta\sigma_{\mathrm{sign}}^2 \log n}{k\mu^2}}.$$

Replacing the rounding condition $1/2$ by 1 therefore changes the sufficient aggregation bound from

$$k \ge \frac{12C\eta\sigma_{\mathrm{sign}}^2 \log n}{\mu^2}$$

to

$$k > \frac{3C\eta\sigma_{\mathrm{sign}}^2 \log n}{\mu^2}.$$

This factor-four change concerns aggregation and parity-constrained rounding. The overall budget must also satisfy Theorem 4.8 and give H_k full column rank.

Noisy parity. Noisy parity can still recover Category III cases missed by ordinary rounding: for $\|\widehat{\varepsilon}\|_\infty < 1$, recovering $\overline{f}$ reduces to the structured LPN system $\overline{\mathbf{y}}_{0,k} = H_k\mathbf{b}_0 + \mathbf{e}_{0,k} \pmod 2$ with secret $\mathbf{b}_0$. At independent flip rate $p_{\mathrm{par}} < 1/2$, $e_{0,k}[\ell] \sim \mathrm{Bernoulli}(p_{\mathrm{par}})$ and $\mathbb{E}[\|\mathbf{e}_{0,k}\|_0] = p_{\mathrm{par}}kn$. Thus smaller p_{par} gives less LPN noise. Solving for $\mathbf{b}_0$ reveals $\overline{f}$ in its first n coordinates and enables parity-constrained rounding to recover f.