Instantiating HAWK with a Remarkable Lattice: E_8 Beyond the Hypercube

Maximilian Adam

Imperial College London, London, UK
max@maximilianadam.com

Abstract. HAWK is a hash-and-sign lattice signature scheme built on the module Lattice Isomorphism Problem, and the only lattice-based candidate that advanced to the third round of the NIST process for additional post-quantum signatures. Its signing routine samples a short vector in a coset of the integer lattice $\mathbb{Z}^{2n}$. We study what changes when this hidden lattice is replaced by direct sums of the remarkable lattice E_8.

We construct HAWK-E_8-CM, a variant over $R_n = \mathbb{Z}[X]/(X^n + 1)$ with $n \in \{256, 512, 1024\}$ that embeds $n/4$ copies of $2E_8$ into HAWK's module through a fixed basis matrix P_n. HAWK's coefficientwise modulo 2 interface cannot address the resulting cosets, so the coset labels are read in P_n coordinates and the public key becomes the modified form $Q_{E_8} = B^\star S_n B$ with $S_n = P_n^\star P_n$. We prove completeness, derive the determinant corrected verification and reconstruction formulas, and determine a public isometry group of order $4n$ that a one-more forgery assumption must quotient by. The recent Galois key recovery attack on HAWK carries over. We prove that its cocycle lattice for HAWK-E_8-CM is $\sqrt{n/4}\,(\mathbb{Z} \oplus \sqrt{2}\,\mathbb{Z}^{n/4-1} \oplus D_6^{\oplus n/8})$, with HAWK's rank, determinant and minimum and the automorphism as its unique shortest vector, and that the descent reaches a HAWK-E_8-CM key of half the degree without reducing the $(2E_8)^{\oplus n/8}$ kernel lattice. Only the attack's provable half dimension reduction does not apply as stated.

A coset matched Construction A sampler with integer tables generated offline meets HAWK's specified precision criterion: per signature, its statistical distance from the ideal is at most $2^{-65.3}$ and its Rényi loss factor at most 1.125, against $2^{-65.2}$ and 1.167 for HAWK's own sampler. In a comparison where only the lattice differs (same precision, sampler technique, randomness, threading, build and fixed-clock core), HAWK-E_8-CM signs 2.33 to 2.50 times slower than HAWK under `-O2` and 1.78 to 1.83 times slower under `-O3 -march=native`, has the higher signing latency with every sampler technique and thread count, and verifies 1.24 to 1.37 times slower. Its uncompressed signatures take 1,038 / 2,072 / 4,136 bytes. The E_8 structure therefore gives neither a signing speed-up nor protection against the Galois attack, and we discuss the size and rank trade-offs where a remarkable lattice may still pay off.

Keywords: Post-quantum signatures · Lattice Isomorphism Problem · HAWK · E_8 lattice · Module LIP · Cyclotomic fields · Discrete Gaussian sampling · HAWK-E_8-CM (CM = Coset Matched)

1 Introduction

The migration to post-quantum public key cryptography is under way. On 13 August 2024 NIST published its first three post-quantum standards, two of which are lattice schemes, among them the signature standard ML-DSA (FIPS 204) [15]. For digital signatures, NIST also ran a separate call for additional schemes, seeking designs based on assumptions other than those already selected. HAWK [3,7] is the only lattice-based candidate that advanced to the third round of that process [1].

HAWK follows the hash-and-sign template of Gentry, Peikert and Vaikuntanathan [12], in the style of Falcon [18], but replaces the NTRU trapdoor with the Lattice Isomorphism Problem (LIP) of Ducas and van Woerden [8], in its module version [7]. The secret key is a hidden rotation of a simple lattice; in HAWK that lattice is $\mathbb{Z}^{2n}$. Because sampling in $\mathbb{Z}^{2n}$ factors into independent one-dimensional discrete Gaussians, signing needs only two small fixed tables and integer arithmetic. This keeps HAWK fast and compact: its signatures are 249 / 555 / 1,221 bytes at its three parameter sets [3].

1.1 Motivation

The integer lattice $\mathbb{Z}^{2n}$ is the simplest possible choice of hidden lattice. The LIP line of work proposes remarkable lattices, with better packing and decoding properties, as a source of better cryptographic trade-offs [8], and earlier work studied how a remarkable lattice such as the Leech lattice can be decoded for cryptographic use [16]. The E_8 lattice is the densest lattice packing in dimension eight [21] and, up to scaling, arises from the $[8, 4, 4]$ extended Hamming code, equivalently the first order Reed–Muller code $\mathrm{RM}(1, 3)$, through Construction A [5], with fast quantizing and decoding algorithms [4]. Since the HAWK degrees are powers of two, $n/4$ copies of an eight dimensional lattice fit its module exactly. This raises three questions, which this paper answers in turn: can HAWK's module and signing interface host a direct sum of E_8 lattices at all, what does the change do to the public key and to the security formulation, and does the lattice change the cost of signing?

An earlier version of this work hoped for faster signing, because the direct sum splits the sampler into independent eight dimensional blocks. That hope does not survive a fair comparison. HAWK's coordinates are independent too, so every form of parallelism open to the E_8 sampler is open to HAWK's, and an E_8 block costs a component choice on top of the eight one-dimensional draws that a block of $\mathbb{Z}^8$ needs. We therefore measure the two schemes with every factor other than the lattice held equal, and find HAWK-E_8-CM slower. The places where a remarkable lattice may still pay off are different ones: a denser packing sampled at a smaller radius could shorten signatures, trading time for size, and realising E_8 inside a higher rank module over a smaller ring would give a finer choice of lattice rank against lattice reduction (section 9).

1.2 Contributions

1. We construct the E_8 block module inside HAWK's ambient module: a congruence defined O_8 submodule $M \subset O_8^2$ with underlying lattice $2E_8$ and basis matrix P, extended by scalars to $M_n = P_n R_n^2 \cong (2E_8)^{\oplus n/4}$ over $R_n = \mathbb{Z}[X]/(X^n+1)$. We prove that HAWK's coefficientwise reduction modulo 2 sees only $n/4$ of the $2n$ parity bits of M_n, and that the internal quotient $M_n/2M_n$, read in P_n coordinates, is the correct coset interface (section 3).
2. We define HAWK-E_8-CM, whose public key is the modified form $Q_{E_8} = B^\star S_n B$ with fixed weight $S_n = P_n^\star P_n$ and determinant $\Delta_n = 48 + 32(Y - Y^3)$. We prove completeness, derive the determinant corrected completion of squares verification formula and a compressed reconstruction rule, bound the public form, and prove that three NTT primes make its verifier sound, where HAWK's range argument works with two for HAWK but cannot for HAWK-E_8-CM (sections 4 and 7).
3. We restate key recovery and forgery for the fixed form class $[S_n]$, and construct the determinant twisted symplectic isometry $\omega_{E_8,Q}$ of Q_{E_8}. We prove it is integral, bijective and norm preserving, determine the public isometry group $G_{E_8,Q}$ of order $4n$ it generates with the scalar rotations, and give the resulting one-more short vector formulation. We then show that the recent key recovery attack on HAWK [20] carries over to HAWK-E_8-CM: we determine the isometry class of its cocycle lattice, prove that the automorphism is the unique shortest vector there and that the descent lattice is $\sqrt{n}\,(2E_8)^{\oplus n/8}$, show that the attack's τ-tower never reduces that lattice, and recover keys at $n \leq 64$ with LLL and short enumeration (sections 5 and 5.4).
4. We give a coset matched Construction A sampler over the sixteen $\mathrm{RM}(1,3)$ labelled components of each block coset, with integer tables generated offline, in three variants that we also build for HAWK, and audit the distance of every variant of both schemes from the ideal discrete Gaussian with one tool, by statistical distance and by Rényi divergence (section 6).
5. We compare the two schemes with only the lattice differing: same sampler precision and technique, same randomness construction, worker pool and thread counts, same signing interface and build, on pinned cores at a fixed clock. At HAWK's specified precision on one core, HAWK-E_8-CM signs 2.33 to 2.50 times slower under `-O2` and 1.78 to 1.83 times slower under `-O3 -march=native`; it is slower with every sampler variant and thread count, key generation costs 0.93 to 1.03 times HAWK's, and verification is 1.24 to 1.37 times slower (section 8).

The paper's main points are therefore the construction, the isometry analysis and the attack analysis of section 5.4, which settles whether leaving the hypercube helps against HAWK's recent key recovery attack: it does not. The Galois cocycle survives, the cocycle lattice keeps HAWK's rank, determinant and minimum, and the descent reduces HAWK-E_8-CM to itself at half the degree without touching the E_8 structure. Only the attack's proof that SVP in dimension $n/2+1$ suffices is lost, because the cocycle lattice leaves the near hypercubic

class that proof covers; heuristically the attack costs the same. We are explicit about the remaining limitations: the sampler and signing path are not fully constant time, signatures and public keys are uncompressed, there is no AVX2 path, and no complete security reduction is given for the modified public form class (section 9).

2 Preliminaries

We keep this section short and reuse the notation of the HAWK specification [3] wherever possible.

2.1 Rings, Forms and Norms

Let n be a power of two and let $R_n = \mathbb{Z}[X]/(X^n + 1)$ be the ring of integers of the $2n$-th cyclotomic field. For $f \in R_n$ we write $f^\star$ for the adjoint under the involution $X \mapsto X^{-1}$, and for a matrix B over R_n we write $B^\star$ for the conjugate transpose. We use the normalized trace inner product $\langle f, g\rangle = \frac{1}{n}\operatorname{Tr}(f^\star g)$, under which the trace form agrees with the usual Euclidean inner product on coefficient vectors. A Hermitian form is a matrix Q over R_n with $Q^\star = Q$ that is positive definite as a real quadratic form; for a basis B the associated form is $Q = B^\star B$, and $\|w\|_Q^2 = \langle w, Qw\rangle$.

2.2 HAWK

We follow the names of the specification's high level algorithm sketches [3]. Key generation (HawkKeyGen) samples $f, g \in R_n$ with coefficients from a centred binomial distribution $\mathrm{Bin}(\eta)$, completes them with NTRUSolve to a basis $B = \left(\begin{smallmatrix} f & F \\ g & G \end{smallmatrix}\right) \in GL_2(R_n)$ with $fG - gF = 1$, and publishes the form $Q = B^\star B$, with entries q_{00}, q_{01}, q_{11}; the secret key is (B, hpub) with $\mathsf{hpub} = H(Q)$. The hidden lattice is $\mathbb{Z}^{2n}$. Signing (HawkSign) hashes $M = H(m\|\mathsf{hpub})$ and a fresh salt to a target $h = H(M\|\mathsf{salt}) \in (R_n/2R_n)^2$, computes $t = Bh \bmod 2$, samples $x \leftarrow D_{2\mathbb{Z}^{2n}+t,\,2\sigma_{\mathrm{sign}}}$, one discrete Gaussian on $2\mathbb{Z}$ or $2\mathbb{Z}+1$ per coordinate, restarts if $\|x\|^2 > 4\sigma_{\mathrm{verify}}^2 \cdot 2n$, and sets $w = B^{-1}x$, so that $w \equiv h \pmod 2$ and $\|w\|_Q = \|x\|$. If sym-break(w) fails, w is replaced by $-w$; the signature is the salt and s_1 of $s = (h - w)/2$, and the verifier rebuilds s_0 (Decompress). Verification (HawkVerify) recomputes h, sets $w = h - 2s$ and accepts if sym-break(w) holds and $\|w\|_Q^2 \le 4\sigma_{\mathrm{verify}}^2 \cdot 2n$.

2.3 Discrete Gaussians and Smoothing

For $\sigma > 0$ we use the standard deviation convention

$$\rho_\sigma(x) = \exp\Big(-\frac{\|x\|^2}{2\sigma^2}\Big), \qquad s = \sqrt{2\pi}\,\sigma \tag{1}$$

where s is the width used in the HAWK tables. The discrete Gaussian on a coset $c + \Lambda$ assigns to each point a probability proportional to its ρ_σ weight. The smoothing parameter $\eta_\varepsilon(\Lambda)$ is the smallest width above which the discrete Gaussian on any coset of Λ is within ε of uniform over the cosets; sampling a signing coset at or above the smoothing parameter of its period lattice is the standard requirement for signatures not to leak the secret basis [2,7,12].

2.4 The E_8 Lattice and Construction A

The E_8 lattice is the unique even unimodular lattice in dimension eight. It is the densest lattice packing in dimension eight [21], with kissing number 240 and minimum norm 2 in the standard scaling [5], where

$$E_8 = \Big\{ y \in \mathbb{Z}^8 \cup \big(\mathbb{Z} + \tfrac{1}{2}\big)^8 : \ \textstyle\sum_i y_i \in 2\mathbb{Z} \Big\}. \tag{2}$$

Construction A maps a binary linear code $C \subseteq \mathbb{F}_2^m$ to the lattice $\{x \in \mathbb{Z}^m : x \bmod 2 \in C\}$. Applying Construction A to the $[8, 4, 4]$ extended Hamming code, which is the first order Reed–Muller code $\mathrm{RM}(1, 3)$, yields a scaling of E_8 [5]. This coding theoretic description is what our sampler uses to label cosets.

3 The E_8 Module and its Coset Interface

We now construct the E_8 block module inside HAWK's ambient module and resolve the modulo 2 interface between the two. Both steps are needed before the scheme in section 4 can be stated.

3.1 The Congruence Module and its Basis

Let $K_8 = \mathbb{Q}(\zeta_8)$ with ring of integers $O_8 = \mathbb{Z}[\zeta_8]$, and write an element of O_8^2 through its eight integer coefficients in the basis $1, \zeta_8, \zeta_8^2, \zeta_8^3$ of each component. The normalized trace form on O_8^2 is then the Euclidean inner product on these coefficient vectors, as for R_n in section 2. The starting object is the congruence defined subset $M \subset O_8^2$ of [11],

$$M = \Big\{ \Big(\textstyle\sum_{i=0}^{3} x_i \zeta_8^i, \ \sum_{i=0}^{3} x_{i+4} \zeta_8^i \Big) \ \Big| \ \sum_{i=0}^{7} x_i \equiv 2x_0 \equiv \cdots \equiv 2x_7 \pmod 4 \Big\} \tag{3}$$

Lemma 1 (The congruence module). *M is an O_8 submodule of O_8^2, and its underlying $\mathbb{Z}$-lattice is isometric to $2E_8$.*

Proof. Let $S = \sum_i x_i$. Since $2x_i \equiv 2x_j$ modulo 4 means $x_i \equiv x_j$ modulo 2, all eight coefficients share one parity. The condition is closed under addition and negation, so it suffices to check stability under multiplication by ζ_8, which maps each block $(x_0, x_1, x_2, x_3) \mapsto (-x_3, x_0, x_1, x_2)$ and so $x \mapsto y = (-x_3, x_0, x_1, x_2, -x_7, x_4, x_5, x_6)$. Then $\sum_i y_i = S - 2(x_3 + x_7) \equiv S \pmod 4$,

since $x_3 + x_7$ is even. Each y_i is $\pm x_j$ for some j, so $2y_i \equiv 2x_j \equiv S$ modulo 4. Hence M is an O_8 submodule.

For the lattice, set $y = x/2$. If all x_i are even then $y \in \mathbb{Z}^8$, and if all are odd then $y \in (\mathbb{Z} + \frac{1}{2})^8$. In both cases S is even, and $S \equiv 2x_j \pmod 4$ becomes $\sum_i y_i \equiv x_j \pmod 2$. So $\frac{1}{2}M$ is the lattice Γ of $y \in \mathbb{Z}^8$ with even sum and $y \in (\mathbb{Z} + \frac{1}{2})^8$ with odd sum. The isometry $\varphi(y_0, \ldots, y_7) = (-y_0, y_1, \ldots, y_7)$ changes $\sum_i y_i$ by $-2y_0$, which is even on $\mathbb{Z}^8$ and odd on $(\mathbb{Z} + \frac{1}{2})^8$, so $\varphi(\Gamma) = E_8$ in the form eq. (2), and $M = 2\Gamma \cong 2E_8$. □

Lemma 2 (A free basis). *$M = P\,O_8^2$ is free of rank 2 over O_8 with basis matrix*

$$P = \begin{pmatrix} 2(1+\zeta_8) & 1 - \zeta_8 + \zeta_8^2 + \zeta_8^3 \\ 0 & 1 + \zeta_8 + \zeta_8^2 + \zeta_8^3 \end{pmatrix} \tag{4}$$

Proof. The two columns of P have the coefficient vectors $(2,2,0,0,0,0,0,0)$ and $(1,-1,1,1,1,1,1,1)$, which satisfy eq. (3), so $L := P\,O_8^2 \subseteq M$ by lemma 1. As $\mathbb{Z}$-lattices, $O_8^2 \cong \mathbb{Z}^8$ has covolume 1, so L has covolume $[O_8^2 : L] = |N_{K_8/\mathbb{Q}}(\det P)|$. Pairing complex conjugate embeddings, $N(1+\zeta_8) = (2+\sqrt{2})(2-\sqrt{2}) = 2$ and $N(1-\zeta_8) = 2$, so $N(2(1+\zeta_8)) = 2^4 \cdot 2 = 32$; and $1 + \zeta_8 + \zeta_8^2 + \zeta_8^3 = 2/(1-\zeta_8)$ has norm $2^4/2 = 8$. Hence L has covolume 256, which is also the covolume of $M \cong 2E_8$, since E_8 is unimodular and scaling by 2 in dimension eight multiplies the covolume by 2^8. A sublattice of equal covolume is the whole lattice, so $L = M$. □

3.2 Extension of Scalars into HAWK's Ring

Fix $n \in \{256, 512, 1024\}$, set $k = n/4$ and $Y = X^k$. Then $Y^4 + 1 = X^n + 1 = 0$ in R_n, so $\zeta_8 \mapsto Y$ embeds O_8 into R_n. Substituting Y for ζ_8 in eq. (4) gives

$$P_n = \begin{pmatrix} 2(1+Y) & 1 - Y + Y^2 + Y^3 \\ 0 & 1 + Y + Y^2 + Y^3 \end{pmatrix} \in M_2(R_n), \qquad M_n := P_n R_n^2 \subset R_n^2 \tag{5}$$

Lemma 3 (Block decomposition). *The ring R_n is a free O_8 module with basis $1, X, \ldots, X^{k-1}$, and $M_n = R_n \otimes_{O_8} M = \bigoplus_{r=0}^{k-1} X^r M$ is an orthogonal direct sum of k copies of M under the normalized trace form. In particular the underlying $\mathbb{Z}$-lattice of M_n is isometric to $(2E_8)^{\oplus n/4}$.*

Proof. Every exponent $0 \le m < n$ is uniquely $m = qk + r$ with $0 \le q \le 3$ and $0 \le r < k$, so every $f \in R_n$ is uniquely $f = \sum_{r<k} X^r a_r(Y)$ with $a_r \in \mathbb{Z}[T]/(T^4+1) \cong O_8$. Extending scalars along this free module turns $M = PO_8^2$ into $M_n = P_n R_n^2$, and the summand $X^r M$ occupies exactly the coefficient positions $r, r+k, r+2k, r+3k$ of each component. These position sets are disjoint for distinct r, and the normalized trace form is the coefficient inner product, so the sum is orthogonal and each summand is isometric to $M \cong 2E_8$ (lemma 1). □

The determinant of P_n is

$$\delta_n := \det(P_n) = 2(1+Y)(1+Y+Y^2+Y^3) = 4(Y + Y^2 + Y^3) \tag{6}$$

which is not a unit, so M_n is a proper submodule of R_n^2, not the image of an ambient unimodular basis. This is the structural reason why the scheme in section 4 must change HAWK's public form rather than only its sampler.

3.3 Two Different Reductions Modulo 2

HAWK signing is tied to the target $t = Bh \bmod 2$ and to coefficientwise cosets $2R_n^2+t$. To connect this to the E_8 block module, one has to decide what reduction modulo 2 means for M_n, and there are two candidates: the internal quotient $M_n/2M_n$, and ambient coefficientwise reduction of the embedded vector inside R_n^2. These are not the same map.

Lemma 4 (Internal quotient). *Let $\pi_M : M_n \to (R_n/2R_n)^2$ send $P_n z$ to $z \bmod 2$. Then π_M is well defined, $\ker(\pi_M) = 2M_n$, and*

$$M_n/2M_n \;\cong\; (R_n/2R_n)^2, \qquad |M_n/2M_n| = 2^{2n} \tag{7}$$

Moreover the quotient decomposes blockwise as $M_n/2M_n \cong \bigoplus_{r=0}^{k-1} X^r(M/2M)$, so a target class consists of $k = n/4$ independent labels, one 8-bit label in $M/2M \cong (O_8/2O_8)^2$ per $2E_8$ block.

Proof. Since P_n is an R_n basis of M_n, every $x \in M_n$ has a unique expression $x = P_n z$ with $z \in R_n^2$, so π_M is well defined. If $\pi_M(x) = 0$ then $z = 2u$ and $x = 2P_n u \in 2M_n$; conversely every element of $2M_n$ has even coordinates. Hence $\ker(\pi_M) = 2M_n$ and π_M induces the stated isomorphism, whose size is $|(R_n/2R_n)^2| = 2^{2n}$. The block decomposition follows from lemma 3 applied in P_n coordinates. □

The internal quotient therefore has exactly the size of HAWK's parity space. Ambient coefficientwise reduction, by contrast, collapses it: in a single block, eq. (3) forces all eight coefficients of an element of M to share one parity, so reducing them modulo 2 gives only $(0,\dots,0)$ and $(1,\dots,1)$, two classes instead of the 2^8 of $M/2M$. Over the whole module:

Lemma 5 (Ambient reduction loses the labels). *Let $\rho : M_n \to (R_n/2R_n)^2$ be coefficientwise reduction modulo 2. Its image has dimension $n/4$ over $\mathbb{F}_2$, against $2n$ for $(R_n/2R_n)^2$. In particular there is no choice of $x_t \in M_n$ with $x_t \equiv t \pmod{2R_n^2}$ for every $t \in (R_n/2R_n)^2$.*

Proof. Modulo 2, $2(1+Y) \equiv 0$ and $1 - Y + Y^2 + Y^3 \equiv 1 + Y + Y^2 + Y^3 =: q$, so $\rho(P_n(a,b)) = (qb, qb) \bmod 2$. Over $\mathbb{F}_2$, $X^n + 1 = (X+1)^n$ since n is a power of two, so $R_n/2R_n = \mathbb{F}_2[\varepsilon]/(\varepsilon^n)$ with $\varepsilon = X + 1$. Moreover $q = (1+Y)^3$ in characteristic 2 and $1 + Y = 1 + X^k = (1+X)^k$, so $q = \varepsilon^{3k}$. Multiplication by ε^{3k} has image $\varepsilon^{3k}\mathbb{F}_2[\varepsilon]/(\varepsilon^n)$, of dimension $n - 3k = n/4$, and the image of ρ is its diagonal copy. A family (x_t) as stated would make ρ surjective. □

So a sampler replacement inside HAWK's coefficientwise interface is impossible, and no other $\mathbb{Z}$-basis of the same module repairs it: the label must be read in P_n coordinates.

3.4 The Coset Matching Map

HAWK fixes the binary section $s : R_n/2R_n \to R_n$ that lifts a class to its representative with coefficients in $\{0, 1\}$, applied componentwise to $(R_n/2R_n)^2$. For a target class $t = Bh \bmod 2$, define the E_8 signing coset

$$\boxed{\mathcal{C}_t := \pi_M^{-1}(t) = P_n\big(2R_n^2 + s(t)\big) = 2M_n + P_n s(t)} \tag{8}$$

The definition is independent of the lift, since replacing $s(t)$ by $s(t)+2u$ changes the right hand side by $2P_n u \in 2M_n$, and $P_n : 2R_n^2 + s(t) \to \mathcal{C}_t$ is a bijection because P_n is an R_n module isomorphism onto M_n. Blockwise, if $t = \sum_{r=0}^{k-1} X^r \tau_r$ with $\tau_r \in (O_8/2O_8)^2$, then

$$\mathcal{C}_t = \bigoplus_{r=0}^{k-1} X^r\, \mathcal{C}_{\tau_r}^{(8)}, \qquad \mathcal{C}_\tau^{(8)} := 2M + Ps(\tau) \tag{9}$$

An E_8 sampler therefore consumes HAWK's parity data blockwise: one 8-bit coset label per block.

One design choice remains. For $x_E \in \mathcal{C}_t$ and $z = P_n^{-1} x_E$, keeping HAWK's public form $Q = B^\star B$ would require sampling with the pulled back weight $\rho_\sigma(P_n^{-1} x_E)$, an anisotropic Gaussian on the E_8 side. We instead sample with the ordinary Euclidean norm on the E_8 block side, which forces the public form to change: for $x_E = P_n B w$,

$$\|x_E\|^2 = \langle w,\ B^\star P_n^\star P_n B\, w\rangle = \langle w, Q_{E_8}\, w\rangle, \qquad Q_{E_8} := B^\star S_n B, \quad S_n := P_n^\star P_n \tag{10}$$

This is the variant constructed in the next section.

4 The HAWK-E_8-CM Construction

4.1 The Fixed Weight and its Determinant

Write $a = 2(1+Y)$, $b = 1 - Y + Y^2 + Y^3$ and $c = 1 + Y + Y^2 + Y^3$, so that $P_n = \left(\begin{smallmatrix} a & b \\ 0 & c \end{smallmatrix}\right)$. Using $Y^4 = -1$ and $Y^\star = Y^{-1} = -Y^3$,

$$S_n = P_n^\star P_n = \begin{pmatrix} a^\star a & a^\star b \\ b^\star a & b^\star b + c^\star c \end{pmatrix} = \begin{pmatrix} 8 + 4(Y - Y^3) & 4Y^2 \\ -4Y^2 & 8 \end{pmatrix} \tag{11}$$

The off-diagonal entries are adjoints of each other, since $(4Y^2)^\star = 4Y^{-2} = -4Y^2$, so S_n is Hermitian. Its determinant is fixed and, with eq. (6),

$$\Delta_n := \det(S_n) = \delta_n^\star \delta_n = \big(8 + 4(Y - Y^3)\big) \cdot 8 - (4Y^2)(-4Y^2) = 48 + 32(Y - Y^3) \tag{12}$$

This replaces the determinant one relation of HAWK. The change looks small but it propagates into public key derivation, verification, and compressed reconstruction.

4.2 The Uncompressed Scheme

We state the scheme without signature compression where the uncompressed signature carries the full vector $s = (s_0, s_1) \in R_n^2$. This is the clean cryptographic object; compression is an encoding layer on top of it, exactly as in HAWK itself.

Key generation. Generate the secret basis $B = \left(\begin{smallmatrix} f & F \\ g & G \end{smallmatrix}\right) \in GL_2(R_n)$, $fG - gF = 1$, exactly as HawkKeyGen does ($f, g \leftarrow \mathrm{Bin}(\eta)$, then NTRUSolve); the trapdoor equation is unchanged. The public key is derived from $Q_{E_8} = B^\star S_n B$ with entries $\tilde{q}_{00}, \tilde{q}_{01}, \tilde{q}_{10} = \tilde{q}_{01}^\star, \tilde{q}_{11}$. Because B is unimodular,

$$\det(Q_{E_8}) = \det(B^\star)\det(S_n)\det(B) = \Delta_n, \qquad \tilde{q}_{00}\tilde{q}_{11} - \tilde{q}_{01}\tilde{q}_{01}^\star = \Delta_n \tag{13}$$

so $\tilde{q}_{11}$ is determined by $\tilde{q}_{00}$ and $\tilde{q}_{01}$ whenever $\tilde{q}_{00}$ is invertible in the ring used for verification. As in HAWK, the public key therefore stores only $\tilde{q}_{00}$ and $\tilde{q}_{01}$, here uncompressed as 32-bit integers, and the secret key is the key seed, F, G and $\mathsf{hpub} = H(\tilde{q}_{00} \| \tilde{q}_{01})$. In place of HAWK's encoding checks, key generation restarts if a coefficient of Q_{E_8} exceeds the key generation limits of section 4.5 or if $\tilde{q}_{00}$ is not invertible modulo one of the verifier's primes (section 7).

Signing. As in HawkSign, compute $M = H(m \| \mathsf{hpub})$, draw a fresh salt per attempt and hash $h = H(M \| \mathsf{salt}) \in (R_n/2R_n)^2$; then:

$$t = Bh \bmod 2 \tag{14}$$

$$x_E \leftarrow \text{sample in } \mathcal{C}_t = 2M_n + P_n s(t) \text{ under the Euclidean } E_8 \text{ norm} \tag{15}$$

$$z = P_n^{-1} x_E \in 2R_n^2 + s(t) \tag{16}$$

$$w = B^{-1} z \qquad B^{-1} = \begin{pmatrix} G & -F \\ -g & f \end{pmatrix} \tag{17}$$

$$s = (h - w)/2 \tag{18}$$

The sample is accepted only if $\|x_E\|^2 \le 8n\,(\sigma_{\mathrm{verify}}^{E_8})^2$, otherwise signing restarts. Before output, a sym-break rule replaces w by $-w$ if needed, so a canonical representative of the $\pm$ pair is encoded (our rule keeps the sign that makes the first non-zero coefficient of w_1, then of w_0, positive). The signature is $\mathsf{salt} \,\|\, s_0 \,\|\, s_1$.

Verification. Recompute h from message, salt and public key, reconstruct $w = h - 2s$, check sym-break, and accept if and only if

$$\|w\|_{Q_{E_8}}^2 \le \left\lfloor 8n\,(\sigma_{\mathrm{verify}}^{E_8})^2 \right\rfloor \tag{19}$$

which is HAWK's bound $4\sigma_{\mathrm{verify}}^2 \cdot 2n$ applied to the new norm.

Proposition 1 (Completeness). *If key generation outputs B and $Q_{E_8} = B^\star S_n B$, signing outputs without restarting, and the verifier recomputes the same h, then the verifier accepts, and $s \in R_n^2$.*

Proof. The signer samples $x_E \in \mathcal{C}_t = P_n(2R_n^2 + s(t))$ with $t = Bh \bmod 2$, so $x_E = P_n z$ for some $z \equiv Bh \pmod 2$. Since $B \in GL_2(R_n)$ we have $B^{-1} \in M_2(R_n)$, hence $w = B^{-1}z \equiv h \pmod 2$. The sign flip preserves this congruence, so $s = (h - w)/2 \in R_n^2$ and the signature is well formed. The verifier recovers $w' = h - 2s = w$ exactly, and the symmetry check passes because the signer enforced the canonical choice. Finally $z = Bw$ gives $x_E = P_n Bw$ and

$$\|x_E\|^2 = \langle P_n Bw, P_n Bw\rangle = \langle w, B^\star P_n^\star P_n B\, w\rangle = \|w\|^2_{Q_{E_8}} \tag{20}$$

so the signer's acceptance condition is exactly the verifier's bound. □

4.3 The Determinant Corrected Verification Formula

HAWK's verifier avoids the naive matrix product through a completion of squares in $\tilde{q}_{00}, \tilde{q}_{01}$. The same idea works here, with the determinant term changing from 1 to Δ_n.

Proposition 2. *Let Q_{E_8} have entries as above with $\det(Q_{E_8}) = \Delta_n$, let $w = (w_0, w_1) \in R_n^2$, and define over the fraction field*

$$d = \frac{w_1}{\tilde{q}_{00}}, \qquad e = w_0 + \tilde{q}_{01} d \tag{21}$$

Then

$$\boxed{n\,\|w\|^2_{Q_{E_8}} = \mathrm{Tr}\big(\tilde{q}_{00}\, ee^\star + \Delta_n\, d\, w_1^\star\big)} \tag{22}$$

Proof. Expanding the Hermitian form gives $w^\star Q_{E_8} w = \tilde{q}_{00} w_0 w_0^\star + \tilde{q}_{01} w_1 w_0^\star + \tilde{q}_{01}^\star w_0 w_1^\star + \tilde{q}_{11} w_1 w_1^\star$. Since $w_1 = \tilde{q}_{00} d$ and $\tilde{q}_{00}^\star = \tilde{q}_{00}$, we have $w_1^\star = d^\star \tilde{q}_{00}$, and expanding the completed square,

$$\tilde{q}_{00} ee^\star = \tilde{q}_{00} w_0 w_0^\star + \tilde{q}_{01} w_1 w_0^\star + \tilde{q}_{01}^\star w_0 w_1^\star + \tilde{q}_{00}\tilde{q}_{01}\tilde{q}_{01}^\star dd^\star \tag{23}$$

The missing term is $\tilde{q}_{11} w_1 w_1^\star = \tilde{q}_{00}^2 \tilde{q}_{11} dd^\star$, and the deficit $\tilde{q}_{00}^2\tilde{q}_{11} - \tilde{q}_{00}\tilde{q}_{01}\tilde{q}_{01}^\star = \tilde{q}_{00}(\tilde{q}_{00}\tilde{q}_{11} - \tilde{q}_{01}\tilde{q}_{01}^\star) = \tilde{q}_{00}\Delta_n$ by eq. (13). Since $dw_1^\star = dd^\star \tilde{q}_{00}$, this deficit equals $\Delta_n dw_1^\star$ up to the trace, giving $\tilde{q}_{00} ee^\star + \Delta_n dw_1^\star = w^\star Q_{E_8} w$. Taking the normalized trace yields eq. (22). □

In HAWK the determinant is 1 and the correction factor is invisible [3], here the verifier multiplies the second term by the fixed element Δ_n. When only s_1 is transmitted, the same completion of squares yields a HAWK style rounding rule that rebuilds s_0 which we derive in section 4.4, together with why HAWK's negligible reconstruction failure analysis does not carry over. We currently only implement the uncompressed scheme.

4.4 Compressed Reconstruction

When only s_1 is stored, verification fixes $w_1 = h_1 - 2s_1$ and must rebuild w_0, as HAWK's Decompress does.

Proposition 3 (Reconstruction rule). *For fixed w_1, the real vector w_0 minimising $\|w\|^2_{Q_{E_8}}$ is $w_0^{\min} = -(\tilde{q}_{01}/\tilde{q}_{00})\, w_1$, the same minimiser as for HAWK with q_{00}, q_{01} replaced by $\tilde{q}_{00}, \tilde{q}_{01}$. The corresponding rounding rule is eq.* (25).

Proof. By proposition 2, for fixed w_1,

$$\|w\|^2_{Q_{E_8}} = \Big\langle w_0 + \tfrac{\tilde{q}_{01}}{\tilde{q}_{00}} w_1,\ \tilde{q}_{00}\big(w_0 + \tfrac{\tilde{q}_{01}}{\tilde{q}_{00}} w_1\big)\Big\rangle + \Big\langle w_1, \tfrac{\Delta_n}{\tilde{q}_{00}} w_1\Big\rangle \tag{24}$$

The second term does not involve w_0, so over the reals the minimum sits where the completed square vanishes, at $w_0^{\min} = -(\tilde{q}_{01}/\tilde{q}_{00})\, w_1$. Writing $w = h - 2s$ and rounding coefficientwise gives eq. (25). □

The rule keeps HAWK's shape:

$$\hat{s}_0 = \Big\lfloor \frac{h_0}{2} + \frac{\tilde{q}_{01}}{\tilde{q}_{00}}\Big(\frac{h_1}{2} - s_1\Big)\Big\rceil \tag{25}$$

with coefficientwise rounding. Δ_n shifts the value of the norm but not the location of the minimiser. What does not carry over is HAWK's negligible failure analysis: its key generation check on $(1/q_{00})[0]$ against β_0 and its fixed point error budget give reconstruction failure below 2^{-105} and 2^{-315} for HAWK-512 and HAWK-1024 [3], and the analogues for $\tilde{q}_{00}^{-1}$ would have to be re-derived. We therefore implement only the uncompressed scheme and leave compression as future work.

4.5 Public Form Scaling

Expanding $Q_{E_8} = B^\star S_n B$ entrywise,

$$\begin{aligned}
\tilde{q}_{00} &= 8(ff^\star + gg^\star) + 4(Y - Y^3)ff^\star + 4Y^2(f^\star g - fg^\star)\\
\tilde{q}_{01} &= 8(Ff^\star + Gg^\star) + 4(Y - Y^3)Ff^\star + 4Y^2(Gf^\star - Fg^\star)\\
\tilde{q}_{11} &= 8(FF^\star + GG^\star) + 4(Y - Y^3)FF^\star + 4Y^2(F^\star G - FG^\star)
\end{aligned} \tag{26}$$

With HAWK's coefficient bounds $|f_i|, |g_i| \le \eta$ and $|F_i|, |G_i| \le 127$ [3], this yields

$$|\tilde{q}_{00}[i]| \le 32n\eta^2, \qquad |\tilde{q}_{01}[i]| \le 32n\eta \cdot 127, \qquad |\tilde{q}_{11}[i]| \le 32n \cdot 127^2 \tag{27}$$

For $n \le 1024$ and $\eta \le 8$ these stay below half of the 31-bit NTT primes $p_1 = 2147473409$ and $p_2 = 2147389441$ used by the HAWK reference code, so every coefficient of the public form fits one prime, as in HAWK. The norm computed by the verifier is a different matter: it needs three primes where HAWK needs two (proposition 5).

In practice the form is far smaller than eq. (27). Over 10^5 keys per degree, the largest coefficients have 13, 15, 17 bits ($\tilde{q}_{00}[0]$), 12, 13, 15 bits ($\tilde{q}_{00}[u]$, $u > 0$), 14, 16, 17 bits ($\tilde{q}_{01}$) and 18, 21, 23 bits ($\tilde{q}_{11}[0]$) at $n = 256, 512, 1024$, consistent with the ten keys of table 3. As HAWK does with its own form, key generation limits every coefficient, the constant terms apart: to 14/12/15/19/18, 16/14/16/21/20

and 18/16/18/24/23 bits for $\tilde{q}_{00}[0]$, $\tilde{q}_{00}[u]$, $\tilde{q}_{01}$, $\tilde{q}_{11}[0]$ and $\tilde{q}_{11}[u]$, one bit above the maxima over 10^4 keys. None of the 10^5 keys per degree is rejected. The corresponding HAWK limits are 9 to 10 bits for q_{00}, 11 to 14 for q_{01} and 13 to 17 for q_{11}, so the Q_{E_8} coefficients are 2^3 to 2^7 times larger. The scale factor is set by S_n:

Lemma 6 (Spectrum of S_n). *Under the four complex embeddings $Y \mapsto y \in \{\zeta_8, \zeta_8^3, \zeta_8^5, \zeta_8^7\}$, the eigenvalues of the embedded Hermitian 2×2 block of S_n are $\lambda_\pm = 8+2r\pm 2\sqrt{6}$ with $r = y - y^3 \in \{\pm\sqrt{2}\}$, numerically 0.2726, 5.9294, 10.0706 and 15.7274. Hence P_n has singular values between about 0.5221 and 3.9658, and*

$$0.2726\,\|v\|^2 \;\le\; \langle v, S_n v\rangle \;\le\; 15.7274\,\|v\|^2 \tag{28}$$

Proof. By eq. (11) the embedded block is $\left(\begin{smallmatrix} 8+4r & 4y^2 \\ 4y^2 & 8 \end{smallmatrix}\right)$ with $r = y - y^3$ real, since $y^3 = -\bar{y}$ when $y^4 = -1$, and $r = 2\,\mathrm{Re}(y) = \pm\sqrt{2}$. Its trace is $16 + 4r$ and its determinant $8(8 + 4r) - 16 = 48 + 32r$, so the eigenvalues are $8 + 2r \pm \sqrt{(8 + 2r)^2 - 48 - 32r} = 8 + 2r \pm \sqrt{4r^2 + 16} = 8 + 2r \pm 2\sqrt{6}$. They are the squared singular values of P_n since $S_n = P_n^\star P_n$, and eq. (28) holds blockwise for every embedding. □

The average eigenvalue is 8, so under roughly isotropic secret coefficients the public form is about 8 times, and at worst about 16 times, the size of the corresponding HAWK form: a budget of 3 to 4 extra bits per public coefficient for a future compressed key. In the same direction, $\tilde{q}_{00}^{-1} \approx \frac{1}{8} q_{00}^{-1}$, so the compression parameter β_0 should initially scale down by about 8.

A useful side effect appears on the security side. HAWK's specification notes that its published omSVP reduction is vacuous for the deployed parameter sets because the public basis vector e_1 is already short enough [3]. Here $\|e_1\|^2_{Q_{E_8}} = \tilde{q}_{00}[0]$, and with the heuristic $\tilde{q}_{00}[0] \approx 8 q_{00}[0] \approx 16 n \sigma^2_{\mathrm{krsec}}$ the vector e_1 falls under the threshold $8n\,(\sigma^{E_8}_{\mathrm{verify}})^2$ only when $\sigma^{E_8}_{\mathrm{verify}} \gtrsim \sqrt{2}\,\sigma_{\mathrm{krsec}}$, a much stricter condition than the one that makes the HAWK reduction vacuous. This is a scaling argument, not a proof, but it suggests the modified form is less exposed to the vacuity mechanism HAWK identifies for the identity form.

5 Public Isometries and Security Formulation

We now state the hardness problems for HAWK-E_8-CM and prove the main structural result of the security analysis: the determinant twisted symplectic isometry of Q_{E_8} and the public isometry group it generates. We stress that this is a formulation and a structural theorem, not a full security proof, section 9 lists what remains open.

5.1 Problems in the Fixed Class $[S_n]$

In HAWK, the public key lives in the equivalence class of the identity form and direct key recovery is rank 2 search module LIP in that class [3,7]. In HAWK-

E_8-CM the public key lives in the fixed class $[S_n]$, and direct key recovery is

$$\text{wc-smLIP}^{S_n}_{K,2}: \quad \text{given } Q_{E_8} \in [S_n],\ \text{find } U \in GL_2(R_n) \text{ with } Q_{E_8} = U^\star S_n U \tag{29}$$

For honest keys $U = B$. Making the class explicit matters as decisional LIP instances are only meaningful between lattices whose efficiently computable invariants agree [8], search to distinguish reductions for direct sum lattices treat the base lattice as part of the fixed problem [11], and the genus and special genus of HAWK like Hermitian lattices [13], hull attacks on LIP and module LIP [6,19], and rank 2 module LIP cryptanalysis over totally real fields [14] all show that the auxiliary structure of the instantiation is part of the attack surface. HAWK-E_8-CM should therefore be analysed as a HAWK like scheme over the class $[S_n]$. The most direct instance is the recent key recovery attack on HAWK [20], which uses a Galois automorphism of R_n together with the integral structure of the identity class; section 5.4 shows that the attack carries over to $[S_n]$.

On the signing side, a valid forgery yields $w = h - 2s$ with $w \equiv h \pmod 2$, the symmetry condition, and $\|w\|^2_{Q_{E_8}} \le 8n\,(\sigma^{E_8}_{\text{verify}})^2$. Equivalently, with $x_E = P_n B w$ we have $Bw \equiv Bh = t \pmod 2$ and $x_E \in 2M_n + P_n s(t)$: a forgery is a short vector in the correctly matched E_8 block coset. The relevant assumption is therefore a one-more short vector problem in the class $[S_n]$, and the delicate part is the group of trivial public isometries that the game must quotient by. Missing publicly computable isometries is non trivial as for HAWK itself an overlooked automorphism quadratically speeds up key recovery and effectively halves security bits [10]. Multiplication by powers of X remains public, since $X^\star X = 1$ and S_n commutes with scalar multiplication. HAWK's determinant one symplectic formula, however, does not carry over verbatim, because $\det(S_n) = \Delta_n \neq 1$. We now show the symplectic symmetry survives in a determinant twisted form.

5.2 The Determinant Twisted Symplectic Isometry

Recall $\delta_n = \det(P_n) = 4(Y + Y^2 + Y^3)$ from eq. (6) and $\Delta_n = \delta_n^\star \delta_n$. Write $\overline{(\cdot)}$ for componentwise adjoint and let $J_2 = \left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right)$. The transported symplectic matrix is

$$\Omega_n := P_n^{-1} J_2\, \overline{P_n} = \begin{pmatrix} -1 + Y - Y^3 & -2Y + 2Y^2 - 2Y^3 \\ Y + Y^3 & -1 + Y - Y^3 \end{pmatrix} \in M_2(R_n) \tag{30}$$

where the point is that the entries are integral although P_n^{-1} appears in the definition.

Theorem 1 (Public isometry group of Q_{E_8}). *Let $Q_{E_8} = B^\star S_n B$ with $B \in GL_2(R_n)$, $\det(B) = 1$, and define the conjugate linear map*

$$\boxed{\omega_{E_8,Q}(w) := \delta_n^\star\, Q_{E_8}^{-1} J_2\, \overline{w}} \tag{31}$$

Then:

1. *$\omega_{E_8,Q}$ is publicly computable from Q_{E_8}, maps R_n^2 bijectively to itself, and preserves the public norm: $\|\omega_{E_8,Q}(w)\|_{Q_{E_8}} = \|w\|_{Q_{E_8}}$ for all $w \in R_n^2$.*
2. *$\omega_{E_8,Q}^2 = -1$ and $\omega_{E_8,Q}(X^i w) = X^{-i}\omega_{E_8,Q}(w)$.*
3. *The group generated by the scalar rotations and $\omega_{E_8,Q}$ is*

$$G_{E_8,Q} = \langle X \cdot I_2,\ \omega_{E_8,Q} \rangle = \{\, X^i,\ X^i\omega_{E_8,Q} : 0 \le i < 2n \,\} \tag{32}$$

   of order $4n$, with presentation $\langle r, \omega \mid r^{2n} = 1,\ \omega^2 = r^n,\ \omega r \omega^{-1} = r^{-1}\rangle$; modulo the centre $\{\pm 1\}$ it acts dihedrally on rotation orbits.
4. *The scalar public isometries are exactly the powers of X.*

Proof. Set $C := P_n B$, so $Q_{E_8} = C^\star C$ and $\det(C) = \det(P_n)\det(B) = \delta_n$. On the C side the map $z \mapsto J_2\overline{z}$ preserves the Euclidean norm. Pulling it back to w coordinates gives $w \mapsto C^{-1}J_2\overline{C}\,\overline{w}$. A direct 2×2 adjugate computation gives $C^{-1}J_2\overline{C} = \delta_n^\star\, Q_{E_8}^{-1} J_2$, which is eq. (31). Although $Q_{E_8}^{-1}$ is taken over the fraction field, integrality follows from

$$C^{-1}J_2\overline{C} = B^{-1}P_n^{-1}J_2\overline{P_n}\,\overline{B} = B^{-1}\Omega_n\overline{B}, \qquad B^{-1},\ \Omega_n,\ \overline{B} \in M_2(R_n) \tag{33}$$

so $\omega_{E_8,Q}$ maps R_n^2 into itself, and it is bijective because it is conjugate to the bijection $z \mapsto J_2\overline{z}$. Norm preservation follows from $C\,\omega_{E_8,Q}(w) = J_2\overline{Cw}$, hence $\|C\,\omega_{E_8,Q}(w)\| = \|Cw\|$, which is the statement in the Q_{E_8} norm.

For (2), $\Omega_n\overline{\Omega_n} = -I_2$ by direct computation, which conjugates to $\omega_{E_8,Q}^2 = -1$, the commutation relation holds because $\omega_{E_8,Q}$ is conjugate linear and $X^\star = X^{-1}$. The relations $X^{2n} = 1$, $\omega_{E_8,Q}^2 = X^n = -1$ and $\omega_{E_8,Q}X\omega_{E_8,Q}^{-1} = X^{-1}$ give (3).

For (4), a scalar $\alpha \in R_n^\times$ preserving every Hermitian form in this setting satisfies $\alpha^\star\alpha = 1$, so $|\sigma(\alpha)| = 1$ under every complex embedding. By Kronecker's theorem such an algebraic integer is a root of unity, and the roots of unity in the power of two cyclotomic ring are exactly $\{1, X, \ldots, X^{2n-1}\}$. □

This is the same HAWK style orbit structure, with the symplectic generator replaced by its determinant twisted, E_8 preconditioned version, and with the square of the conjugate linear generator being the central sign [3,10].

5.3 The One-More Formulation

The public orbit removed from the forgery game is

$$G_{E_8,Q} \cdot w = \{\, X^i w,\ X^i\omega_{E_8,Q}(w) : 0 \le i < 2n \,\} \tag{34}$$

The resulting assumption then reads: given Q_{E_8}, samples $w_1, \ldots, w_N \leftarrow D_{Q_{E_8},\sigma}$ and the public group $G_{E_8,Q}$, it is hard to output a short $w^\star \in R_n^2$ with

$$w^\star \notin \bigcup_{j=1}^{N} G_{E_8,Q} \cdot w_j \tag{35}$$

Theorem 1 constructs the uniform public isometries forced by the scalar rotations and the twisted symplectic map, so the one-more game must quotient at least by $G_{E_8,Q}$. This does not prove that this group exhausts all efficiently computable public isometries for generic keys, nor rule out accidental extra automorphisms for special sampled keys and treating $G_{E_8,Q}$ as the complete public orbit is an additional assumption [10]. Nor does it prove signature security: the rest of HAWK's reduction, including the mod 2 bad event and fixed coset counting arguments, remains to be redone for the class $[S_n]$ and the twisted map. Finally, $G_{E_8,Q}$ contains only isometries of a single public form. The Galois automorphism $X \mapsto -X$ instead relates Q_{E_8} to a second public form, and this is what the attack of section 5.4 exploits.

5.4 The Galois Key Recovery Attack

The recent key recovery attack against HAWK [20] has three steps. Write $K_n = \mathbb{Q}[X]/(X^n+1)$, $\tau : X \mapsto -X$, σ for the composite of τ and the involution $\star$, adj for the adjugate of a 2×2 matrix, $\mathrm{Tr} = \mathrm{Tr}_{K_n/\mathbb{Q}}$, $F \subset K_n$ for the fixed field of τ and $\star$, of degree $n/4$, and $O' = R_n \cap K_n^\tau = \mathbb{Z}[X^2]$, a copy of $R_{n/2}$. For a public form Q the cocycle lattice is

$$\Lambda(Q) = \left\{ Y \in M_2(R_n) :\ \tau(Y) = \operatorname{adj} Y,\ QY = \sigma(Y)^T \tau(Q) \right\} \tag{36}$$

with the form $Q^{(\tau)}(Y) = \mathrm{Tr}_{F/\mathbb{Q}}(\det Y)$; it is the integer kernel of a linear system whose coefficients are computed from Q. First, for HAWK, $\Lambda(Q)$ has rank n, contains $V_\tau = B^{-1}\tau(B)$ with $Q^{(\tau)}(V_\tau) = n/4$ as a shortest vector, and is isometric to $\sqrt{n/4}\,(\mathbb{Z}^{n/2+1} \oplus \sqrt{2}\,\mathbb{Z}^{n/2-1})$ [20, Prop. 4.5]. Second, Ducas's block reduction, which [20, Thm. 5.1] analyses for this near hypercubic class, returns its $n+2$ shortest vectors with calls to exact SVP in dimension $n/2+1$, and a parity test selects $\pm V_\tau$. Third, the descent of van Gent and Pulles [10] turns V_τ into a key, either by one more run of Ducas's algorithm on the kernel lattice $\{x \in R_n^2 : \tau(x) = V_\tau^{-1} x\} \cong \sqrt{n}\,\mathbb{Z}^n$ or, in the attack's implementation, through a τ-tower that turns V_τ into a HAWK key of degree $n/2$ by algebra alone and recurses [20, Lemma D.4]. We redo each step for Q_{E_8}.

Lemma 7 (The cocycle survives). *Since $n/4$ is even, $\tau(Y) = Y$, so $\tau(P_n) = P_n$ and $\tau(S_n) = S_n$. For every key $B \in SL_2(R_n)$:*

1. *$V_\tau^\star\, Q_{E_8}\, V_\tau = \tau(Q_{E_8})$, $\det V_\tau = 1$ and $V_\tau \in \Lambda(Q_{E_8})$;*
2. *$\varphi_\tau : Y \mapsto BY\tau(B)^{-1}$ maps $\Lambda(Q_{E_8})$ onto $\Lambda(S_n)$, preserves* det *(hence $Q^{(\tau)}$), and sends V_τ to I.*

Proof. $\tau(Y) = \tau(X)^{n/4} = (-1)^{n/4} Y = Y$, and P_n, S_n are polynomials in Y. Since τ commutes with $\star$, $\tau(Q_{E_8}) = \tau(B)^\star S_n\, \tau(B)$, and

$$V_\tau^\star Q_{E_8} V_\tau = \tau(B)^\star B^{-\star} B^\star S_n B\, B^{-1}\tau(B) = \tau(B)^\star S_n\, \tau(B) = \tau(Q_{E_8})$$

The rest is the computation of [20, Lemmas 4.1 and 4.2] with S_n inserted. $\det V_\tau = \tau(\det B)/\det B = 1$, so $\tau(V_\tau) = V_\tau^{-1} = \operatorname{adj} V_\tau$. With $\tau(B^\star) = \sigma(B)^T$

and $\tau(S_n) = S_n$, $\sigma(V_\tau)^T \tau(Q_{E_8}) = B^\star \sigma(B)^{-T} \sigma(B)^T S_n \, \tau(B) = Q_{E_8} V_\tau$. For $W = \varphi_\tau(Y)$ one has $\sigma(W)^T = B^{-\star} \sigma(Y)^T \sigma(B)^T$, so the second constraint for Q_{E_8} and Y, $B^\star S_n B Y = \sigma(Y)^T \sigma(B)^T S_n \, \tau(B)$, is equivalent to $S_n W = \sigma(W)^T S_n$, the second constraint for S_n and W. The first constraint and det are preserved because $B, \tau(B) \in SL_2(R_n)$, exactly as in [20, Lemma 4.2]. □

The cocycle lattice of Q_{E_8} is therefore public, and its isometry class does not depend on the key. It is not the class of HAWK:

Proposition 4 (The cocycle lattice of HAWK-E_8-CM). *Let $n \geq 16$, and let L_{12} be the lattice $P\,\mathfrak{sl}_2(O_8)P^{-1}$, with P from eq. (4), under the sum of the squared coefficient norms of the four entries. Then*

$$\big(\Lambda(Q_{E_8}), Q^{(\tau)}\big) \;\cong\; \sqrt{n/4}\,\big(\mathbb{Z} \oplus \sqrt{2}\,\mathbb{Z}^{n/4-1} \oplus L_{12}^{\oplus n/16}\big) \tag{37}$$

and:

1. $\det L_{12} = 16$, *so $\Lambda(Q_{E_8})$ has rank n and the determinant $(n/4)^n 2^{n/2-1}$ of HAWK's cocycle lattice;*
2. *L_{12} has minimum at least 2, so the minimum $n/4$ of $\Lambda(Q_{E_8})$ is attained only at $\pm V_\tau$, and every other nonzero vector has $Q^{(\tau)} \geq n/2$;*
3. *by exact computation, $L_{12} \cong D_6 \oplus D_6$, so $\Lambda(Q_{E_8}) \cong \sqrt{n/4}\,(\mathbb{Z} \oplus \sqrt{2}\,\mathbb{Z}^{n/4-1} \oplus D_6^{\oplus n/8})$.*

Section B proves eq. (37) and items (1) and (3). The idea is to conjugate by P_n. Since $\tau(P_n) = P_n$, the map $W \mapsto P_n W P_n^{-1}$ carries $\Lambda(S_n)$ onto the lattice of those rational solutions $W(a, b)$ of the constraints for $Q = I$ [20, Sect. 4.3] that stabilise M_n. Splitting W into its X-components over $K_8 = \mathbb{Q}(Y)$ (lemma 3), the even components give HAWK's summand $\sqrt{n/4}\,(\mathbb{Z} \oplus \sqrt{2}\,\mathbb{Z}^{n/4-1})$, and each of the $n/16$ pairs of odd components contributes one copy of L_{12}. Conjugation by P has determinant 1 on $\mathfrak{sl}_2(K_8)$, which gives (1), and (3) is a finite exact computation on a rank 12 Gram matrix. Item (2) is the one place where the structure of E_8 enters. Its proof uses $P^\star P = 4S'$ with $S' = \left(\begin{smallmatrix} 2+Y-Y^3 & Y^2 \\ -Y^2 & 2 \end{smallmatrix}\right)$ (eq. (11)), which lies in $GL_2(O_8)$ because $\det S' = 3 + 2(Y - Y^3) = (1 + Y - Y^3)^2$ is a unit.

Proof (of (2)). Let $W = PZP^{-1} \in L_{12}$ with $Z \in \mathfrak{sl}_2(O_8)$. Then $h = \mathrm{tr}(W^\star W) = \mathrm{tr}(Z^\star S' Z S'^{-1})$ is a totally positive element of $O_8 \cap \mathbb{Q}(\sqrt{2}) = \mathbb{Z}[\sqrt{2}]$, where $\sqrt{2} = Y - Y^3$, and $\|W\|^2 = \frac{1}{4}\,\mathrm{Tr}_{K_8/\mathbb{Q}}(h)$ is its rational part, an integer. Suppose $\|W\|^2 = 1$. The lattice $\Gamma = \frac{1}{2}M \cong E_8$ (lemma 1) is stable under W; the real linear map of W on coefficient vectors has squared Hilbert–Schmidt norm $\mathrm{Tr}_{K_8/\mathbb{Q}}(h) = 4$; and the 240 roots ρ of Γ satisfy $\sum_\rho \rho\rho^T = 60\, I_8$, so $\sum_\rho \|W\rho\|^2 = 240$. Each $W\rho$ lies in the even lattice Γ, so it is 0 or has norm at least 2. As $W \neq 0$, $\ker W$ is 0 or a K_8-line, which meets Γ in a lattice of rank at most 4 and so contains at most 24 roots, the kissing number in dimension four [5]. At least 216 roots then contribute at least 2 each, a contradiction. Hence a vector of $\Lambda(S_n)$ with a nonzero odd part has $Q^{(\tau)} \geq n/2$, and in the even part only $\pm I$ reach $n/4$; φ_τ maps $\pm V_\tau$ to $\pm I$. □

At $n = 8$, below the parameter sets but needed by the descent, the odd components form a single self-paired block, and direct computation gives $\mathbb{Z} \oplus \sqrt{2}\,\mathbb{Z} \oplus D_6$, the value of eq. (37) at $n = 8$. Leaving the hypercube therefore changes the class of the cocycle lattice but not its rank, its determinant or its minimum, and it makes $\pm V_\tau$ the only shortest vectors, where HAWK has $n + 2$; the parity test is no longer needed. What is lost is the provable second step. Equation (37) is not of the form $\mathbb{Z}^p \oplus \sqrt{2}\,\mathbb{Z}^q$, so [20, Thm. 5.1] does not apply as stated, and we know of no proof that SVP in dimension about $n/2$ suffices for this class. (It satisfies $\lambda_1\lambda_1^\vee = 1/\sqrt{2}$, the hypothesis of the framework of Bambury and Nguyen that [20] builds on, but only through the summand $\sqrt{2}\,\mathbb{Z}^{n/4-1}$; once that is split off, $\mathbb{Z} \oplus D_6^{\oplus n/8}$ has $\lambda_1\lambda_1^\vee = 1$.) The heuristic cost is unchanged: the estimate of [20, App. C] for progressive BKZ depends only on the rank, the determinant and the length of the target, which proposition 4 shows are those of HAWK, so it predicts the same block sizes, 205 and 432 at $n = 512$ and 1024, and the same heuristic costs of about $2^{80.8}$ and $2^{146.5}$ gates.

The descent lattice is the one the earlier version of this paper expected:

Lemma 8 (The descent). *Let $n \geq 16$, and for $V = \pm V_\tau$ let $M_V = \{x \in R_n^2 : \tau(x) = V^{-1}x\}$ with $T(x, x') = \mathrm{Tr}(x^\star Q_{E_8} x')$.*

1. *$M_{V_\tau} = B^{-1}O'^2$ and $M_{-V_\tau} = B^{-1}(XO')^2$.*
2. *$(M_{\pm V_\tau}, T) \cong \sqrt{n}\,(2E_8)^{\oplus n/8}$.*
3. *If $x, y \in M_{V_\tau}$ and $U = [x\ y]$ has $\det U = 1$, then $A = BU \in SL_2(O')$ and $U^\star Q_{E_8} U = A^\star S_n A$; under $X^2 \mapsto X$ this is a HAWK-E_8-CM public key of degree $n/2$ with secret basis A.*

The proof, in section B, follows [20, Prop. D.2]: $\tau(x) = \pm V_\tau^{-1}x$ if and only if $\tau(Bx) = \pm Bx$, and for $x = B^{-1}u$, $T(x, x') = n\langle P_n u, P_n u'\rangle$, while $P_n O'^2$ is a copy of $M_{n/2}$.

The attacker never has to reduce this lattice. The step of the τ-tower, [20, Lemma D.4], uses only $V_\tau \equiv I \pmod 2$, $V_\tau\tau(V_\tau) = I$, $M_{V_\tau} = B^{-1}O'^2$ and $\det B = 1$, all of which hold here; the first because τ acts trivially on $R_n/2R_n$. So M_{V_τ} is generated over O' by the columns of $(I+V_\tau)/2$ and $(I-V_\tau)/(2X)$, and a pair (x, y) in it with $\det[x\ y] = 1$, found by a Bézout step in O', is a basis. By lemma 8(3) this gives a HAWK-E_8-CM key of degree $n/2$, whose cocycle lattice is eq. (37) at degree $n/2$, and the recursion continues to degree 8. Its descent leaves the rank 8 lattice $(O_8^2, A^\star SA) \cong 2E_8$, which an attacker solves by listing its 240 minimal vectors. The parity test does not separate V_τ from $-V_\tau$; for $-V_\tau$ the same step yields a basis (x, y) of $B^{-1}(XO')^2$, and $(X^{-1}x, Xy)$ is then a basis of $B^{-1}O'^2$ of determinant 1, so an attacker tries both signs. The result is a $B' \in GL_2(R_n)$ with $B'^\star S_n B' = Q_{E_8}$, which signs in place of B, since the proof of proposition 1 uses only these two properties. Neither the cocycle lattice nor the descent lattice is ever decoded, so the fast E_8 decoder [4], a property of the signer, plays no part on either side.

Experiments. From public keys at $n = 8$, 16 and 32 (five keys each) and $n = 64$ (two), in exact arithmetic, the class certificate of eq. (37), the uniqueness of $\pm V_\tau$

and lemma 8(2) all check out, and the τ-tower recovers an equivalent key every time, with LLL and short enumeration only (section B). The experiments test the structure, not the cost.

Consequently the Galois attack applies to HAWK-E_8-CM. The E_8 base lattice removes neither the cocycle nor the descent. It changes the class of the cocycle lattice, but not the parameters that the heuristic cost depends on, and the shortest vector it makes unique is the one the attack wants. It costs the attacker only the provable half dimension guarantee of [20], until a reduction for the class eq. (37) is found. We do not count that as a security margin: HAWK-E_8-CM at $n \in \{256, 512, 1024\}$ should be assumed to fall to the same heuristic attack as HAWK, at the same estimated cost. The mechanism is general. Every primitive eighth root of unity in K_n is an odd power of $X^{n/4}$ and is fixed by τ when $n \geq 8$, so any instantiation whose fixed weight has entries in $\mathbb{Q}(\zeta_8)$ keeps the cocycle of lemma 7. An O_8 structure cannot avoid this within power-of-two rings, and the attack's own exception, a conductor m with cyclic $(\mathbb{Z}/m)^\times$ [20, Thm. 7.1], excludes ζ_8 altogether, since $8 \mid m$ makes $(\mathbb{Z}/m)^\times$ non-cyclic.

6 The Coset Matched Sampler and its Accuracy

6.1 The Construction A Sampler

Signing must sample $x_E \in \mathcal{C}_t$ under the Euclidean E_8 norm. By eq. (9) the problem splits into $k = n/4$ independent eight-dimensional block problems, one per $2E_8$ copy, and this independence is what later enables the block-parallel signing path. For a block label $\tau \in (O_8/2O_8)^2$, read off the target as in lemma 4, the block sampler returns $z_{\mathrm{blk}} \in 2\mathbb{Z}^8 + \tau$ with weight proportional to

$$\exp\Big(-\frac{\|Pz_{\mathrm{blk}}\|^2}{2\,(\sigma_{\mathrm{sign}}^{E_8})^2}\Big), \qquad x_{\mathrm{blk}} = Pz_{\mathrm{blk}} \in \mathcal{C}_\tau^{(8)} = 2M + Ps(\tau) = P(2\mathbb{Z}^8 + \tau) \quad (38)$$

Internally the sampler writes $z_{\mathrm{blk}} = s(\tau) + 2y$ and decomposes the coset over the sixteen components labelled by the codewords of $\mathrm{RM}(1,3)$. For each component it computes the Gaussian mass under the E_8 side norm, selects a component according to these masses, samples the eight shifted one-dimensional integer coordinates inside the selected component, and reconstructs z_{blk} with a checked inverse. Debug builds verify parity, coset membership and the block norm identity.

Integer tables. Because $\sigma_{\mathrm{sign}}^{E_8}$ is fixed per parameter set, every table (the one-dimensional distributions of the thirteen centre classes, the component distributions of all 256 labels τ and the reconstruction data) is generated offline from exact rational exponents and committed as integer constants. The signing path therefore contains no floating point; we check this by compiling every signing translation unit with `-mgeneral-regs-only` and scanning the timed objects for floating-point instructions.

Variants. We build the sampler in three variants, and the same three for HAWK, so that a comparison pair differs only in the lattice:

- `ref`: 78-bit thresholds in HAWK's draw format and constant-time full table scans. The component is chosen by a masked scan over all 256 labels and their component tables, each coordinate by a masked scan over the tables of the four centre residues, with no early exit. For HAWK, `ref` is the specified sampler. The constant-time property of the E_8 `ref` rests on its structure, checked by counting every loop's iterations for every label and random input, and on code review, not on a tool such as ctgrind.
- `hot64` and `hot32`: 64- or 32-bit thresholds, a variable time scan over the six most likely values and, on a miss, an integer tail table. HAWK's variants use the same skeleton on its two one-dimensional distributions.

Randomness. Every sampler stream of both schemes is bound as HAWK binds its own: SHAKE256 absorbs $H(m\|\mathsf{hpub})$, the key material, the attempt counter and a 40-byte seed from the signing RNG, followed by a domain label. With T workers, worker w samples a contiguous range of blocks (HAWK-E_8-CM) or coordinates (HAWK) from its own stream, so the output depends on T but not on scheduling; a serial call uses one stream and no worker pool.

6.2 Parameter Calibration

The signing width is chosen at or above the smoothing parameter of the period lattice of the fixed signing coset [2,12]. For fixed t the period lattice of $\mathcal{C}_t$ is $2M_n \cong (4E_8)^{\oplus n/4}$. Starting from the kissing number based estimate $\eta_\varepsilon(E_8) \approx \frac{1}{\sqrt{2}}\sqrt{\frac{1}{\pi}\log(240/\varepsilon)}$ of [9], replacing the 240 shortest vectors by the $240\,(n/4) = 60n$ of the product lattice, scaling by the factor 4 of the period lattice, and converting to the σ convention of eq. (1), gives the full signing space estimate

$$\sigma_{\mathrm{smooth}}(n,\varepsilon) \approx \frac{2}{\pi}\sqrt{\log\Big(\frac{60n}{\varepsilon}\Big)} \tag{39}$$

With $\varepsilon = 2^{-32}$ and rounding upward to three decimals this yields $\sigma_{\mathrm{sign}}^{E_8} = 3.592$, 3.631, 3.669 for $n = 256$, 512, 1024. A sweep with a loose verification radius confirmed that these smallest admissible widths already show near maximal Construction A component entropy, low first shell concentration, and no diagnostic failures.

The verification width was then tuned to the smallest radius with retry pressure near 10^{-5}, matching HAWK's behaviour. From the empirical block norm mean μ_{blk} and variance v_{blk} of the sweep, the full norm over $n/4$ blocks was approximated by $\frac{n}{4}\mu_{\mathrm{blk}} + z_{1-p}\sqrt{\frac{n}{4}v_{\mathrm{blk}}}$ with $p = 10^{-5}$ and $z_{1-p} \approx 4.265$, giving

$$\sigma_{\mathrm{verify}}^{E_8} \approx \sqrt{\frac{\frac{n}{4}\mu_{\mathrm{blk}} + 4.265\sqrt{\frac{n}{4}v_{\mathrm{blk}}}}{8n}} \quad\Longrightarrow\quad \sigma_{\mathrm{verify}}^{E_8} = 2.03,\ 1.99,\ 1.95 \tag{40}$$

The validation of section 8.2 confirms these values: over 10^6 signatures per sampler variant and parameter set, HAWK-E_8-CM needs 1.000008 to 1.000039 attempts per signature. Table 1 collects the resulting prototype parameters.

Table 1. Prototype HAWK-E_8-CM parameters used for the experiments.

Parameter	$n = 256$	$n = 512$	$n = 1024$
Centered binomial parameter η	2	4	8
$\sigma_{\text{sign}}^{E_8}$	3.592	3.631	3.669
$\sigma_{\text{verify}}^{E_8}$	2.030	1.990	1.950
Bound $\lfloor 8n\,(\sigma_{\text{verify}}^{E_8})^2 \rfloor$	8439	16220	31150
Salt length (bytes)	14	24	40
Hashed public key (bytes)	16	32	64
Signature, uncompressed (bytes)	1,038	2,072	4,136
Public key $\tilde{q}_{00} \| \tilde{q}_{01}$ (bytes)	2,048	4,096	8,192
Private key (bytes)	544	1,080	2,152
Verifier NTT primes	3	3	3

6.3 Accuracy of the Discrete Gaussian

All sampler tables of both schemes are generated offline from exact rational exponents and stored as integer constants, so signing is integer only and the implemented probability of every outcome is an exact rational with a power-of-two denominator. We audit three sampler variants per scheme with one tool and one set of measures: `ref`, with 78-bit thresholds and a constant time full table scan (for HAWK the specified sampler [3]), and `hot64` and `hot32`, with 64- and 32-bit thresholds and a variable time table of the six most likely values followed by an integer tail table. All tables round as HAWK's specified tables do (floored tail masses, so extreme values are rounded down). The ideal distributions are the discrete Gaussians with the same parameters, evaluated to 220 decimal digits.

Measures. For an implemented distribution P and the ideal Q we report the statistical distance $\Delta(P, Q) = \frac{1}{2}\sum_x |P(x) - Q(x)|$ and the Rényi divergence of order a as HAWK defines and uses it [3, eq. (5) and §6.5],

$$R_a(P \,\|\, Q) = \Big(\sum_{x \in \mathrm{Supp}(P)} \frac{P(x)^a}{Q(x)^{a-1}} \Big)^{1/(a-1)}, \tag{41}$$

in the specification's direction, implemented against ideal. A HAWK signature samples $2n$ independent coordinates, an HAWK-E_8-CM signature $n/4$ independent blocks. The block is in bijection with its component and eight product coordinates, and its weight in eq. (38) is the product of the eight one-dimensional weights, so the block divergence is computed exactly,

$$R_a(P_{\text{blk}} \,\|\, Q_{\text{blk}})^{a-1} = \sum_c P(c)^a Q(c)^{1-a} \prod_{v=1}^{8} R_a(P_{c,v} \,\|\, Q_{c,v})^{a-1},$$

Table 2. Distance of each sampler variant from the ideal discrete Gaussian, for one signature at $n = 256$, 512 and 1024: $\log_2$ of the union bound on the statistical distance, and the loss factor L_a of eq. (42) at its optimal order. A dash marks a loss factor above 2^{10^8}, i.e. no useful bound.

			$\log_2 \Delta_{\mathrm{sig}}$			$\min_a L_a$		
Scheme	Variant	Bits	256	512	1024	256	512	1024
HAWK	`ref`	78	−67.6	−66.8	−65.2	1.0001	1.060	1.167
HAWK-E_8-CM	`ref`	78	−67.1	−66.2	−65.3	1.0000	1.029	1.125
HAWK	`hot64`	64	−54.6	−53.4	−52.9	1.0002	$1.8 \cdot 10^9$	$1.9 \cdot 10^{15}$
HAWK-E_8-CM	`hot64`	64	−54.1	−53.0	−51.7	1.0000	1.004	1.006
HAWK	`hot32`	32	−22.8	−21.4	−20.2	7.898	–	–
HAWK-E_8-CM	`hot32`	32	−21.9	−20.7	−20.0	16.85	–	–

and the statistical distance of a block is bounded by that of the component plus the eight coordinates. Per signature we take every coordinate at its worse parity and every block at its worst label τ, with R_a multiplicative over independent draws and Δ bounded by the union bound. Over q_s signatures the probability preservation argument of HAWK [3, eqs. (57)–(58)], [17] loses at most the factor

$$L_a = R_a(P_{\mathrm{sig}} \,\|\, Q_{\mathrm{sig}})^{q_s}\, 2^{\lambda/(a-1)}, \tag{42}$$

which we minimise over integer orders $a \in [2, 2^{20}]$, with $\lambda = 64, 128, 256$ and $q_s = 2^{32}, 2^{64}, 2^{64}$ for $n = 256, 512, 1024$ [3], for both schemes.

Support. The tables' ideal reference keeps the integers within $20s + 32$ of each centre, s the one-dimensional width. The mass outside is below 2^{-1488} for every distribution of both schemes, so it changes no reported value: the smallest is 2^{-147}. Every table also truncates the ideal support, so the reverse divergence $R_a(Q \,\|\, P)$ is infinite for every variant of both schemes, as it is for HAWK's specified tables.

Validation. The audit reproduces the specification's figures for HAWK `ref` [3, §6.5]. The per-coefficient divergences at order 513 are $1 + 2^{-79.2}$ and $1 + 2^{-78.6}$ for HAWK-512 and HAWK-1024 (specified: below $1 + 2^{-78}$). The loss factors are $L_{37286} = 1.0602$ and $L_{4181} = 1.1674$ (specified: 1.07 and 1.17), and these orders are the minimisers of eq. (42). The exact moments of every table agree with an independent computation in the C test suite.

Reading. By statistical distance the two schemes are equally accurate at equal threshold width: per signature they differ by at most 1.2 bits (table 2). Where HAWK-E_8-CM is slightly worse, the cause is its $9n/4$ draws per signature against HAWK's $2n$ and its 16-entry component tables. At high orders the divergence is set by the floored deficit of the extreme entries, of order 2^{-B} times the mass of their table. Both `ref` variants thus meet [3, eq. (6)] at every n, with per-draw

$R_{513} - 1$ of $2^{-79.2}$ to $2^{-78.3}$ (HAWK) and $2^{-80.0}$ to $2^{-77.3}$ (HAWK-E_8-CM). By the specification's per-coefficient criterion $R_{513} - 1 < 2^{-78}$, HAWK `ref` passes at every n and HAWK-E_8-CM `ref` at $n = 256$ and 512 ($2^{-78.5}$ and $2^{-80.0}$), but not at $n = 1024$ ($2^{-77.3}$, from two floored extreme entries). Hot tail thresholds are relative to the tail mass, 2^{-22} to 2^{-16} for HAWK-E_8-CM but 2^{-9} to 2^{-5} for the wider HAWK distributions, so HAWK-E_8-CM `hot64` meets every criterion ($R_{513} - 1 \leq 2^{-83.6}$) and HAWK `hot64` ($2^{-71.0}$ to $2^{-69.1}$) meets eq. (6) only at $n = 256$; neither `hot32` does. The timing tables of section 8 mark these eq. (6) results per row. An earlier rule of our generator rounded the extreme entry up, and single entries then decided these figures (loss 8.3 for HAWK-E_8-CM `ref` at $n = 512$). The tables' integer construction also replaces the `double` construction error that dominated the heuristic estimate of our earlier prototype.

7 Implementation

The implementation is a C extension of the HAWK reference code, gated behind an experimental build flag and kept apart from it: the HAWK sources, formats and tests are byte-identical to upstream, which the build checks. Its interface mirrors HAWK's: key generation, signing from an encoded private key or from a prepared one (the counterpart of HAWK's decoded private key), public key decoding and verification. Key generation calls HAWK's own NTRU key generation and then computes Q_{E_8} by an NTT modulo one prime, as HAWK computes Q. The E_8 additions comprise a target coset unit over $\mathbb{F}_2$ for eq. (14), reconstruction, public form and verification arithmetic, the sampler variants of section 6.1 on their generated integer tables, and a codec for the uncompressed formats of table 1.

A signing call follows eqs. (14) to (18) directly: derive the salt with HAWK's construction, hash to the target, compute the coset labels over $\mathbb{F}_2$, run the $n/4$ block samplers, decide the norm test of eq. (19) on the sampled norm alone, and only then perform the single reconstruction, sym-break and encoding. Products over $\mathbb{F}_2$ are bit packed, with a per key comb table turning each negacyclic product into a branchless window scan. Products in R_n use one negacyclic NTT unit in the Montgomery domain, and the reconstruction $w = B^{-1}z$ uses a single 31-bit prime, justified by a proven centred bound on every reconstructed coefficient (section A). The prepared key holds the NTT transforms of the basis and the comb tables, so signatures under a prepared key skip that setup, as HAWK's decoded key skips its key expansion.

The $n/4$ block samplers run serially, or on a worker pool over contiguous block ranges with one bound stream per worker (section 6.1). The same pool and stream split serve the parallel HAWK variants of section 8.

7.1 Verification and its Primes

The verifier evaluates eq. (22) as HAWK's verifier evaluates its own completion of squares: per prime, the NTT of w_1 and the auto-adjoint NTT of $\tilde{q}_{00}$, a batch

inversion, d, e and the half-trace, with twiddles, Montgomery constants and the NTT of Δ_n generated offline, an integer bound and an early reject after the first prime. It differs from HAWK's only where the lattice does: the factor Δ_n, w_0 taken from the signature rather than reconstructed, and the number of primes. Write $V = \frac{n}{2}\|w\|^2_{Q_{E_8}}$ for the half-trace the verifier computes modulo each prime, and $p_0p_1p_2 = 2^{93.00}$ for the product of its three primes.

Proposition 5 (Three primes suffice). *For every key within the key generation limits of section 4.5 and every signature with* 16*-bit coefficients,* $0 \le V < p_0p_1p_2$*. Hence the verifier, which accepts if its three residues agree on a value* r *with* $\frac{n}{2} \mid r$ *and* $r/\frac{n}{2}$ *within the bound, returns the mathematical verdict.*

Proof. For polynomials a, b, q with $|a| \le A$, $|b| \le B'$ coefficientwise and $L = \max_{k\neq 0}|q_k|$, $|[a^\star qb]_0| \le nAB'|q_0| + n(n-1)AB'L$, since the n pairs of equal indices meet the constant term. Expanding $\|w\|^2_{Q_{E_8}}$ into its four terms, with $|w_0|, |w_1| \le W = 2^{16}+1$ (as $w = h - 2s$ for 16-bit s) and the limits $L = 2^{\text{bits}} - 1$, gives $V \le 2^{73.34}$, $2^{78.19}$ and $2^{84.10}$ at $n = 256$, 512, 1024, below $2^{93.00}$. As $V \ge 0$ because Q_{E_8} is positive definite, agreement of the residues determines V. The early reject is sound because the acceptance threshold $\frac{n}{2}\lfloor 8n(\sigma^{E_8}_{\text{verify}})^2 \rfloor$ is below 2^{24}, so below every prime. □

Two primes ($p_0p_1 = 2^{62.00}$, HAWK's) cannot be made sound by the same argument. With the key limits alone, fitting 2^{62} would need $|w| \le 1285$, 239 and 30 at the three degrees, while honest signatures reach $|w_0| = 2{,}010$, 5,945 and 15,646. Even HAWK-style signature limits one bit above the honest maxima of s_0 and s_1 give $V \le 2^{62.18}$, $2^{69.59}$ and $2^{77.00}$. The cause is the scaling of section 4.5: the Q_{E_8} coefficients are 2^3 to 2^7 times HAWK's. The third prime is therefore a cost of the E_8 public form and stays in our comparison, while wider primes, open to both schemes, are used by neither.

8 Evaluation

We compare HAWK and HAWK-E_8-CM so that the only intended difference is the lattice. Every other factor that affects speed is held equal for both schemes, or varied identically and reported: sampler precision and technique, the randomness construction, threading, the signing interface and its accounting, the build, and the machine state. A claim about the lattice rests only on rows that are fair in this sense; the remaining rows are labelled as implementation context.

8.1 Methodology

Machine. All timings come from one AMD Ryzen 9 9950X3D (16 cores in two CCDs of 8) under Arch Linux, kernel 7.1.5, with GCC 16.1.1. SMT is off, the `amd-pstate` driver runs in passive mode with the `performance` governor, and boost is off, so every core runs at the 4.29 GHz base clock (busy loops measured

4.286 to 4.289 GHz on the pinned cores; in active mode the driver ignores the boost setting). Unless stated otherwise every process is pinned to the 8 cores of the CCD with the 96 MiB V-cache L3. A preflight check refuses to start a run unless all of this holds.

Builds. One compiler and one flag set build every object of a timed binary, HAWK's and HAWK-E_8-CM's alike, with debug checks and profiling off. We report two flag sets, which move the ratios considerably: `O2` is `-O2`, as upstream HAWK builds, and `O3native` is `-O3 -march=native -fno-strict-aliasing`. The last flag is needed because the upstream HAWK reference code accesses one buffer through both 64-bit and 32-bit pointers in its $\mathbb{F}_2$ multiplication, undefined behaviour that `-O3 -march=native` miscompiles, so that its signatures fail verification; it costs at most 1.5% and applies to both schemes.

Measure and schedule. The primary measure is core cycles from the performance counters; wall-clock (TSC) cycles serve for parallel latency, and at the base clock the two agree within 2% on one core. Per configuration we sign with 10 keys × 1,000 signatures, the same messages and signing-RNG streams for both schemes, with the configurations interleaved round-robin per batch so that drift spreads over all of them, after one warm-up signature each. A run repeats this in a fresh process; the estimate is the mean over 5 runs of the per-run median, and ratios carry 95% confidence intervals from a hierarchical bootstrap that resamples runs and then signatures. Every signature is verified, outside the timed region.

Accounting. All-in times the whole signing call from the encoded private key, salt derivation, every attempt and encoding included. *Prepared* starts from HAWK's decoded private key and from HAWK-E_8-CM's prepared basis, whose costs appear in the stage profile. HAWK's signature is compressed and HAWK-E_8-CM's is not; both encodings are inside the timed call.

Pairs. The fair pairs are `ref`/`ref` (78-bit constant-time scans, HAWK's specified precision), `hot64`/`hot64` and `hot32`/`hot32` (section 6.1). In parallel runs both schemes use the same worker pool and stream split, and HAWK's `ref` becomes `ref-T`: its specified table and scan on the per-worker streams.

Acceptance. The headline run set signed and verified 49,200,000 signatures without a failure, every attempt count matched the validation, the per-run medians of all 444 configurations agree within 3% (within 1.2% except the AVX2 context rows), and every one of 2,400,000 timed verifications accepted while all 6000 tampered signatures were rejected. Three implementation artefacts found on this machine were removed before these runs, without changing any output: data-dependent branches in HAWK-E_8-CM's modular arithmetic under `O3native`, and stack staging buffers in the constant-time scans of HAWK-E_8-CM `ref` and HAWK `ref-T` that 4K-aliased their table loads.

Table 3. Maximum absolute coefficients of $Q_{E_8} = B^\star S_n B$, ten keys and 10,000 trials per parameter size.

n	$\max\lvert\tilde{q}_{00}\rvert$	bits	$\max\lvert\tilde{q}_{01}\rvert$	bits	$\max\lvert\tilde{q}_{11}\rvert$	bits
256	4696	13	6040	13	118,336	17
512	18,376	15	26,116	15	811,040	20
1024	69,088	17	95,160	17	6,139,912	23

8.2 Correctness and Validation

Before timing, every sampler variant of both schemes signed 10^6 signatures per parameter set with real keys, each verified under the encoded and the decoded public key ($2.1 \cdot 10^7$ signatures, some of them with 4 worker threads), with no signing or verification failure; a debug build reran every HAWK-E_8-CM signature with its block coset, parity and norm checks and gave identical signatures. The samples were also tested against the exact implemented distributions of section 6.3: 9774 chi-square and G-tests on every one-dimensional table, every component and block norm distribution per label τ, the product coordinates and the signature norm, none rejected at 10^{-3} after Holm's correction, while all 10035 negative controls were rejected. HAWK-E_8-CM needs 1.000008 to 1.000039 attempts per signature, against 1.1792 to 1.1796 for HAWK at $n = 256$ and at most 1.000007 above, and the mean signature norm agrees with the continuous approximation $2n(\sigma_{\text{sign}}^{E_8})^2$ to a relative 10^{-4}. The validation tools and their results are part of the artifact `https://github.com/Maximilian-Adam/hawk_e8`.

8.3 Sizes

Table 3 reports the largest absolute coefficients of the public form over ten keys per parameter set; section 4.5 gives the maxima over 10^5 keys and the key generation limits.

The uncompressed signature $\mathsf{salt}\|s_0\|s_1$ takes $\text{salt} + 4n$ bytes, 1,038 / 2,072 / 4,136, which is 3.4 to 4.2 times HAWK's compressed 249 / 555 / 1,221 bytes. The uncompressed public key $\tilde{q}_{00}\|\tilde{q}_{01}$ takes $8n$ bytes, 2,048 / 4,096 / 8,192, which is 3.4 to 4.6 times HAWK's 450 / 1,024 / 2,440 bytes (table 12 in section C lists every object). By section 4.5 a future compressed public key should cost roughly 3 to 4 bits per coefficient more than HAWK's.

8.4 Signing on One Core

Table 4 is the headline comparison: both schemes at HAWK's specified precision on one core. HAWK-E_8-CM is slower in every row, all-in by 2.33 to 2.50 times under `O2` and by 1.78 to 1.83 times under `O3native`, and prepared by 1.95 to 2.00 and 1.41 to 1.45 times. The flag set matters because the E_8 constant-time scan gains far more from `-O3 -march=native` than HAWK's does. With the faster variable time technique in both schemes (table 5) the gap narrows but does

Table 4. Signing on one core at the as-specified precision: HAWK `ref` against E8-CM `ref` (78-bit thresholds, constant-time full table scan). kcycles: thousands of core cycles per signature on one core, 5 runs (setup, accounting and estimator in section 8.1). ratio: E8-CM/HAWK; every 95% bootstrap CI half-width is at most 0.011. eq. (6): whether HAWK/E8-CM meets eq. (6) of the HAWK specification (table 2). random bytes: drawn per signature, attempts included. Attempts per signature: HAWK 1.182 to 1.185 at $n = 256$, at most 1.000 above; E8-CM at most 1.000.

			random bytes		`O2` (kcycles)			`O3native` (kcycles)		
n	acct	eq. (6)	HAWK	E8-CM	HAWK	E8-CM	ratio	HAWK	E8-CM	ratio
256	all-in	✓/✓	6,133	5,814	123.0	307.5	2.50	110.0	201.0	1.83
	prep.		6,117	5,814	126.8	253.2	2.00	112.8	159.5	1.41
512	all-in	✓/✓	10,304	11,584	257.5	626.1	2.43	219.7	401.2	1.83
	prep.		10,304	11,584	262.3	511.3	1.95	223.3	316.5	1.42
1024	all-in	✓/✓	20,560	23,120	565.3	1,319.2	2.33	468.8	836.8	1.78
	prep.		20,560	23,120	539.1	1,051.1	1.95	445.2	644.6	1.45

Table 5. Signing on one core with the same variable-time hot-table sampler in both schemes, at 64 bits (`hot64`) and 32 bits (`hot32`). kcycles: thousands of core cycles per signature on one core, 5 runs (setup, accounting and estimator in section 8.1). ratio: E8-CM/HAWK; every 95% bootstrap CI half-width is at most 0.004. eq. (6): whether HAWK/E8-CM meets eq. (6) of the HAWK specification (table 2). random bytes: drawn per signature, attempts included. Attempts per signature: HAWK 1.170 to 1.179 at $n = 256$, at most 1.000 above; E8-CM at most 1.000.

			random bytes		`O2` (kcycles)			`O3native` (kcycles)		
n	acct	eq. (6)	HAWK	E8-CM	HAWK	E8-CM	ratio	HAWK	E8-CM	ratio
`hot64`, 64-bit thresholds										
256	all-in	✓/✓	4,898	4,662	123.8	198.8	1.61	111.2	167.3	1.50
	prep.		4,908	4,662	126.8	144.7	1.14	113.1	125.8	1.11
512	all-in	–/✓	8,423	9,280	258.2	410.0	1.59	223.1	334.6	1.50
	prep.		8,423	9,280	262.3	295.1	1.13	225.8	250.0	1.11
1024	all-in	–/✓	16,832	18,512	559.6	886.7	1.58	468.2	701.0	1.50
	prep.		16,832	18,512	537.7	618.4	1.15	448.7	509.2	1.13
`hot32`, 32-bit thresholds										
256	all-in	–/–	2,479	2,358	101.4	174.1	1.72	89.3	143.9	1.61
	prep.		2,468	2,358	104.4	120.3	1.15	91.3	102.5	1.12
512	all-in	–/–	4,244	4,672	213.2	361.1	1.69	179.9	287.9	1.60
	prep.		4,244	4,672	217.3	246.2	1.13	182.6	203.4	1.11
1024	all-in	–/–	8,456	9,296	470.5	789.0	1.68	383.0	607.4	1.59
	prep.		8,456	9,296	448.5	520.9	1.16	363.5	415.6	1.14

Table 6. The samplers alone on one core: cycles and random bytes per 8 coordinates ($2n$ coordinates per call; the 40-byte seed included), median per call, mean over 5 runs. bits: threshold width; `ref`: constant-time full table scan; `hot64`, `hot32`: variable-time hot table. HAWK `ref-T`: HAWK's `ref` on E8-CM's per-worker stream split (the HAWK half of the parallel pairs).

			`O2` cycles			`O3native` cycles			random bytes		
scheme	variant	bits	256	512	1024	256	512	1024	256	512	1024
HAWK	`ref`	78	992	1,003	1,017	982	984	995	80.6	80.3	80.2
HAWK	`ref-T`	78	1,030	1,072	1,060	1,153	1,143	1,127	80.6	80.3	80.2
HAWK	`hot64`	64	1,013	1,015	999	1,008	1,007	988	64.8	65.6	65.6
HAWK	`hot32`	32	659	660	650	663	671	656	32.7	33.0	32.9
E8-CM	`ref`	78	2,986	2,957	2,953	1,650	1,621	1,612	90.6	90.3	90.2
E8-CM	`hot64`	64	1,226	1,209	1,202	1,097	1,068	1,067	72.6	72.3	72.2
E8-CM	`hot32`	32	841	822	817	715	697	691	36.6	36.3	36.2

not close: all-in 1.58 to 1.72 (`O2`) and 1.50 to 1.61 (`O3native`), prepared 1.13 to 1.16 and 1.11 to 1.14. Precision does not explain the gap: within each pair both samplers have the same threshold width, and their statistical distances agree within 1.2 bits (table 2).

Table 6 isolates the samplers. At equal width and technique HAWK's sampler is faster per eight coordinates in every row. Under `O2`, HAWK `ref` takes 992 to 1,017 cycles against 2,953 to 2,986 for HAWK-E_8-CM `ref`, `hot64` 999 to 1,015 against 1,202 to 1,226, and `hot32` 650 to 660 against 817 to 841. Under `O3native` HAWK-E_8-CM `ref` drops to 1,612 to 1,650, against 982 to 995 for HAWK. This is the cost structure to expect: an E_8 block needs a component choice on top of the eight one-dimensional draws that a block of $\mathbb{Z}^8$ needs, and in the constant-time variant that choice is a scan over all 256 labels.

Table 7 shows where the signing time goes. The HAWK-E_8-CM sampler is 56 to 61% of its all-in signing time under `O2` and 49 to 51% under `O3native`, against 46 to 55% and 54 to 60% for HAWK's. HAWK-E_8-CM's key preparation, which builds the NTT and $\mathbb{F}_2$ basis tables, costs 9.4 times HAWK's key expansion at $n = 256$ (60.8 against 6.5 thousand cycles) and 5.0 times at $n = 1024$, under `O2`. Reconstruction costs about the same in both schemes (3 to 9% more for HAWK-E_8-CM under `O2`, 13 to 18% under `O3native`). The sampler and the key preparation together account for most of the difference; the prepared rows, which omit the key preparation, are correspondingly closer. An earlier version of this work attributed most of the E_8 signing time to reconstruction; that profile came from a sampler spread over 8 to 16 threads, and does not hold on one core.

8.5 Parallel Signing and Throughput

Both samplers parallelise: HAWK's $2n$ coordinates are as independent as HAWK-E_8-CM's $n/4$ blocks. With both schemes on the same worker pool and stream

Table 7. Where the signing time goes: stage profile of both `ref` signers on one core, all-in, thousands of TSC cycles per signature, mean over 5 runs (profiling builds: the timers are included, so the totals exceed table 4). HAWK: reconstruction computes s_1, norm checks include sym-break. E8-CM: key preparation builds the prepared basis, reconstruction computes s.

	$n = 256$		$n = 512$		$n = 1024$	
stage	HAWK	E8-CM	HAWK	E8-CM	HAWK	E8-CM
`O2`, thousands of cycles						
key preparation	6.5	60.8	15.5	131.7	59.8	301.2
salt, hashing	5.0	6.6	4.6	7.7	7.1	12.0
target coset	3.0	6.9	6.7	18.0	19.2	58.6
sampler	76.7	188.4	129.1	374.2	260.7	748.6
reconstruction	40.4	44.0	89.9	94.7	195.3	201.6
norm checks	2.2	0.5	3.8	0.5	7.0	0.5
encoding	3.3	0.9	8.1	1.4	15.9	2.5
other	1.3	1.5	1.2	1.5	1.2	1.5
total	138.3	309.6	258.9	629.8	566.1	1,326.7
`O3native`, thousands of cycles						
key preparation	6.5	50.3	14.7	103.7	53.0	229.6
salt, hashing	5.0	6.9	4.7	8.0	7.2	12.6
target coset	2.8	7.0	6.1	16.4	17.3	45.0
sampler	75.6	105.0	125.2	207.3	253.5	417.5
reconstruction	29.7	33.6	59.4	68.1	117.9	139.4
norm checks	2.2	0.5	3.8	0.5	7.0	0.5
encoding	2.5	0.3	6.5	0.3	12.8	0.4
other	1.2	1.4	1.1	1.5	1.1	1.6
total	125.5	205.0	221.6	405.8	469.9	846.5

Table 8. Parallel signing latency, E8-CM/HAWK, both schemes on the same worker pool and stream split with T threads on T cores: ratio of wall-clock (TSC) cycles at T=1 to 8, and CPU, the ratio of core cycles summed over all threads (total CPU per signature) at $T = 8$. HAWK `ref-T`: HAWK's `ref` on the worker streams. Every 95% CI half-width is at most 0.005; cycles in table 15.

		`O2`					`O3native`				
n	acct	T=1	2	4	8	CPU	T=1	2	4	8	CPU
`ref-T/ref`											
256	all-in	2.46	2.28	2.15	2.05	2.32	1.67	1.76	1.83	1.87	1.65
	prep.	1.98	1.67	1.44	1.27	1.95	1.30	1.26	1.21	1.18	1.35
512	all-in	2.36	2.21	2.11	2.04	2.30	1.67	1.77	1.87	1.93	1.67
	prep.	1.90	1.61	1.40	1.25	1.89	1.31	1.26	1.22	1.19	1.34
1024	all-in	2.30	2.15	2.05	1.98	2.27	1.67	1.75	1.83	1.88	1.65
	prep.	1.90	1.63	1.42	1.28	1.89	1.34	1.28	1.25	1.23	1.33
`hot64`											
256	all-in	1.61	1.71	1.79	1.85	1.58	1.51	1.64	1.74	1.82	1.49
	prep.	1.14	1.11	1.09	1.09	1.18	1.11	1.11	1.11	1.12	1.16
512	all-in	1.59	1.70	1.80	1.85	1.51	1.50	1.65	1.79	1.86	1.42
	prep.	1.13	1.10	1.09	1.07	1.10	1.11	1.11	1.12	1.12	1.08
1024	all-in	1.59	1.69	1.77	1.83	1.56	1.50	1.64	1.76	1.84	1.47
	prep.	1.15	1.13	1.12	1.12	1.14	1.13	1.15	1.17	1.18	1.12
`hot32`											
256	all-in	1.72	1.78	1.85	1.88	1.68	1.61	1.71	1.80	1.86	1.58
	prep.	1.15	1.11	1.10	1.09	1.21	1.12	1.11	1.12	1.13	1.19
512	all-in	1.69	1.79	1.85	1.89	1.62	1.60	1.74	1.84	1.91	1.52
	prep.	1.13	1.11	1.09	1.08	1.12	1.11	1.12	1.13	1.13	1.10
1024	all-in	1.68	1.76	1.82	1.85	1.63	1.59	1.72	1.82	1.87	1.54
	prep.	1.16	1.14	1.13	1.12	1.14	1.14	1.16	1.18	1.18	1.12

Table 9. Throughput per core: $P = 8$ single-thread signer processes, one per core, running at once. Signatures per second of signing time, mean over the processes and the 5 runs; ratio E8-CM/HAWK (above 1 favours E8-CM), every 95% CI half-width at most 0.003. *E8-CM ahead: 1.008, 95% CI [1.007, 1.009].

		`O2` (sig/s/core)			`O3native` (sig/s/core)		
n	acct	HAWK	E8-CM	ratio	HAWK	E8-CM	ratio
`ref`							
256	all-in	31,613	13,867	0.44	35,104	21,336	0.61
	prep.	30,411	16,932	0.56	33,859	26,879	0.79
512	all-in	16,668	6,847	0.41	19,536	10,696	0.55
	prep.	16,363	8,384	0.51	19,217	13,547	0.70
1024	all-in	7,606	3,253	0.43	9,160	5,127	0.56
	prep.	7,976	4,083	0.51	9,645	6,653	0.69
`hot64`							
256	all-in	31,638	21,420	0.68	34,892	25,661	0.74
	prep.	30,576	29,713	0.97	33,864	34,139	1.01*
512	all-in	16,633	10,474	0.63	19,243	12,833	0.67
	prep.	16,382	14,556	0.89	19,014	17,170	0.90
1024	all-in	7,672	4,842	0.63	9,160	6,124	0.67
	prep.	7,987	6,943	0.87	9,558	8,432	0.88
`hot32`							
256	all-in	39,257	24,400	0.62	44,107	29,824	0.68
	prep.	37,787	35,692	0.94	42,675	41,926	0.98
512	all-in	20,135	11,882	0.59	23,871	14,913	0.62
	prep.	19,767	17,422	0.88	23,518	21,111	0.90
1024	all-in	9,125	5,438	0.60	11,209	7,064	0.63
	prep.	9,574	8,235	0.86	11,810	10,325	0.87

split (table 8), HAWK-E_8-CM is slower at every thread count, by 1.07 to 2.46 times in wall-clock cycles. At $T = 8$ the ratio under `O2` is 1.98 to 2.05 all-in and 1.25 to 1.28 prepared for `ref-T`/`ref`, and 1.07 to 1.12 prepared for the `hot` pairs; under `O3native` it is 1.87 to 1.93, 1.18 to 1.23 and 1.12 to 1.18. Under `O2` the prepared `ref-T`/`ref` ratio falls with T (1.98 to 1.27 at $n = 256$) while the all-in ratios of the `hot` pairs rise (1.61 to 1.85 for `hot64` at $n = 256$). The total CPU per signature behaves like the serial cost: at $T = 8$ its ratio stays within 0.14 of the pair's $T = 1$ ratio. A separate run set across both CCDs, with its own conditions and a known asymmetry in the data the workers move (table 14 in section C), gives 1.12 to 2.00 at $T = 16$; it is never mixed with the headline tables.

Throughput per core (table 9), with $P = 8$ independent single-thread signers on the eight cores, gives HAWK-E_8-CM 0.41 to 0.98 of HAWK's rate in every row but one, `ref` 0.41 to 0.56 under `O2` and 0.55 to 0.79 under `O3native`. The exception is `hot64` at $n = 256$, prepared, under `O3native`: 34,139 against 33,864 signatures per second per core, a ratio of 1.008 (95% CI $[1.007, 1.009]$), because

Table 10. Key generation on one core: millions of core cycles per key, 5 runs of 1,000 keys on paired seeds. mean: mean of the per-run means (restarts make the cost heavy-tailed, so the mean is the expected cost); median: mean of the per-run medians; ratio: E8-CM/HAWK of the means (E8-CM/AVX2 on the AVX2 row), every 95% paired-bootstrap CI half-width at most 0.013. core: HAWK's NTRU key generation core, which both schemes call; tail: mean minus core on the same keys, in thousands of cycles (HAWK: public-key encoding; E8-CM: public form, limits and invertibility over 3 primes). diverged: keys on which HAWK restarted after the core because its compressed public key failed to encode, which E8-CM's uncompressed key never does; hence the ratio below 1 at $n = 512$ and 1024. †Implementation context (E8-CM has no AVX2 path).

	`O2` (Mcycles)				`O3native` (Mcycles)				
scheme	mean	median	tail	ratio	mean	median	tail	ratio	diverged
$n = 256$									
HAWK	2.59	2.41	25		2.41	2.22	23		
E8-CM	2.67	2.50	102	1.03	2.46	2.28	74	1.02	0.5%
core	2.57	2.39			2.38	2.21			
AVX2†	1.28	1.19	17	2.09	1.21	1.14	16	2.03	
AVX2 core†	1.26	1.18			1.19	1.13			
$n = 512$									
HAWK	9.34	7.88	792		8.54	7.16	723		
E8-CM	8.77	8.04	217	0.94	7.97	7.26	150	0.93	8.5%
core	8.55	7.82			7.82	7.11			
AVX2†	5.47	4.42	468	1.60	5.38	4.34	459	1.48	
AVX2 core†	5.00	4.39			4.92	4.31			
$n = 1024$									
HAWK	49.47	46.30	854		47.14	44.07	819		
E8-CM	49.07	40.77	462	0.99	46.62	38.04	299	0.99	1.6%
core	48.61	40.32			46.32	37.74			
AVX2†	34.19	32.47	607	1.44	34.08	32.41	606	1.37	
AVX2 core†	33.58	26.18			33.48	25.82			

HAWK's `hot64` prepared signer loses about 11% of its one-core rate when eight processes run at once and HAWK-E_8-CM's about none; on one core the same pair is 1.11 times slower for HAWK-E_8-CM (table 5). We draw no conclusion about the lattice from this row.

8.6 Key Generation and Verification

Key generation (table 10) costs HAWK-E_8-CM 1.03, 0.94 and 0.99 times HAWK's under `O2` and 1.02, 0.93, 0.99 under `O3native`. Both schemes run the same NTRU core, which dominates. HAWK-E_8-CM's own tail after it (public form, limits, invertibility over three primes) costs more than HAWK's public key encoding, but HAWK restarts its whole key generation on 0.5, 8.5 and 1.6% of keys when

Table 11. Verification on one core: thousands of core cycles per signature and E8-CM/HAWK ratios (every 95% CI half-width at most 0.024), 5 runs of 10 keys × 1,000 signatures. prep.: the public key decoded outside the timed call (decoding costs HAWK 11.5 to 53.5 and E8-CM 22.0 to 87.4 kcycles). primes: NTT primes of the norm check, the minimum each scheme's range argument proves sound (proposition 5). fast: E8-CM's verifier with HAWK's techniques. †ref: the verifier of an earlier version (tables rebuilt per call, one inversion per coefficient, floating-point bound), context only.

		`O2` kcycles		ratio		`O3native` kcycles		ratio	
verifier	primes	all-in	prep.	all-in	prep.	all-in	prep.	all-in	prep.
$n = 256$									
HAWK	2	115.7	105.9			87.1	78.3		
E8-CM fast	3	158.2	135.4	1.37	1.28	119.1	98.5	1.37	1.26
E8-CM ref†	4	771.1	742.0	6.66	7.01	708.7	688.3	8.14	8.79
$n = 512$									
HAWK	2	247.4	226.2			175.0	157.0		
E8-CM fast	3	331.7	287.5	1.34	1.27	234.9	194.8	1.34	1.24
E8-CM ref†	4	1,560.8	1,519.9	6.31	6.72	1,421.2	1,382.1	8.12	8.80
$n = 1024$									
HAWK	2	530.8	485.0			356.3	312.5		
E8-CM fast	3	697.8	610.8	1.31	1.26	473.2	394.0	1.33	1.26
E8-CM ref†	4	3,201.3	3,120.8	6.03	6.44	2,875.5	2,797.9	8.07	8.95

its compressed public key fails to encode, which HAWK-E_8-CM's uncompressed key never does; this key format effect puts the ratio below 1 at $n = 512$ and 1024.

Verification (table 11) with HAWK's techniques on both sides costs HAWK-E_8-CM 1.26 to 1.37 times HAWK's under `O2` and 1.24 to 1.37 times under `O3native`, with three primes against two (proposition 5). The third prime is a cost of the E_8 public form. The verifier of an earlier version of this work, kept as the context row `ref`, rebuilt its tables on every call and used four primes; it is 6.03 to 7.01 times HAWK's under `O2` and 8.07 to 8.95 times under `O3native`.

8.7 Reading the Results

No fair row supports a speed advantage from the lattice. On one core, in parallel at every thread count, with every sampler technique and in verification, HAWK-E_8-CM is slower; its only lead, 0.8% in one throughput row, comes from how HAWK's signer behaves under concurrent load. The direct sum decomposition $M_n \cong (2E_8)^{\oplus n/4}$ does expose block parallelism, but HAWK's coordinates expose finer parallelism, and per block the E_8 sampler does more work. The upstream AVX2 implementation of HAWK widens the gap further (table 13 in section C), but HAWK-E_8-CM has no AVX2 path, so that table is context only.

9 Limitations and Future Work

Incomplete security argument. HAWK reduces strong unforgeability to search module LIP and a one-more short vector assumption [3,7]. HAWK-E_8-CM moves the published object to the class $[S_n]$, and we only restate the problems and construct the forced public isometries (section 5), so the mod 2 bad event and coset counting arguments must be redone for the new class; we also give no quantum random oracle model proof. On the attack side the answer is negative: the recent Galois key recovery attack carries over to $[S_n]$, heuristically at HAWK's cost (section 5.4), so HAWK-E_8-CM has no security advantage over HAWK against it. Still open are a provable half dimension reduction for the new cocycle class, which would restore the attack's provable bound, recovery experiments beyond $n = 64$, and further structure that the preconditioned class may expose through S_n and P_n beyond the isometries of theorem 1; an overlooked automorphism already speeds up key recovery against HAWK [10].

Side channels. The signing runtime is integer only, on tables generated offline. The `ref` sampler is a constant-time full table scan whose constant-time property rests on its structure, checked by loop counts, and on code review rather than on a tool such as ctgrind; the `hot64` and `hot32` samplers are variable time, and the rest of the signing path is not hardened. There is no AVX2 path, where HAWK ships AVX2 optimised code [3].

Size and format. Signatures and public keys are uncompressed, which is why signatures are 3.4 to 4.2 times HAWK's. We settle the reconstruction rule and the coefficient budget (sections 4.4 and 4.5), but the failure analysis and the encodings remain future work.

Performance. Under a comparison in which only the lattice differs, the E_8 structure makes signing slower, not faster (section 8). Faster implementations of either scheme remain possible, but every technique we measured is open to both.

Future work. The most useful next steps are a remarkable lattice instantiation that the Galois attack does not reach, which cannot use a weight with entries in $\mathbb{Q}(\zeta_8)$ over a power-of-two ring, since every such weight is fixed by τ (section 5.4), and a compressed signature and public key format from eq. (25) and section 4.5. Two directions may change the trade-off in favour of a remarkable lattice, and we present both as open questions rather than results. First, size: a denser packing sampled at a smaller radius, which needs both a small smoothing parameter and a good sampler for the lattice, could shorten signatures. E_8, or the Barnes–Wall family more generally, may improve on HAWK there, at a cost in signing time. Second, rank: instead of $n/4$ copies of E_8 over R_n with $n \geq 256$, one can realise E_8 inside a higher rank module over a smaller ring $\mathbb{Z}[X]/(X^{n'} + 1)$, $n' \geq 8$. This would allow a finer choice of the Euclidean lattice rank against block reduction, which is where a remarkable lattice's resistance to lattice reduction would pay off, provided the Galois attack, which we analysed only for rank 2, does not

extend to such modules. HAWK-E_8-CM is therefore a design study: it shows how a remarkable lattice can be made to fit HAWK's module and signing interface, and what that costs, but neither our measurements nor the attack analysis give a reason to prefer it to HAWK as it stands.

References

1. Alagic, G., Moody, D., Bros, M., Ciadoux, P., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Liu, Y.K., Miller, C., Peralta, R., Perlner, R., Robinson, A., Silberg, H., Smith-Tone, D., Waller, N.: Status report on the second round of the additional digital signature schemes for the NIST post-quantum cryptography standardization process. NIST Internal Report 8610, NIST (2026). https://doi.org/10.6028/NIST.IR.8610
2. Bollauf, M.F., Lie, M., Ling, C.: On Gaussian sampling for q-ary lattices and linear codes with Lee weight. In: Advances in Cryptology – CRYPTO 2025, Part I. Lecture Notes in Computer Science, vol. 16000, pp. 321–352. Springer (2025). https://doi.org/10.1007/978-3-032-01855-7_11
3. Bos, J.W., Bronchain, O., Ducas, L., Fehr, S., Huang, Y.H., Pornin, T., Postlethwaite, E.W., Prest, T., Pulles, L.N., van Woerden, W.: HAWK: Specification document, version 1.1. NIST Post-Quantum Cryptography, Additional Digital Signature Schemes (2025), https://hawk-sign.info
4. Conway, J.H., Sloane, N.J.A.: Fast quantizing and decoding algorithms for lattice quantizers and codes. IEEE Transactions on Information Theory **28**(2), 227–232 (1982). https://doi.org/10.1109/TIT.1982.1056484
5. Conway, J.H., Sloane, N.J.A.: Sphere Packings, Lattices and Groups, Grundlehren der mathematischen Wissenschaften, vol. 290. Springer, 3 edn. (1999). https://doi.org/10.1007/978-1-4757-6568-7
6. Ducas, L., Gibbons, S.: Hull attacks on the lattice isomorphism problem. In: Public-Key Cryptography – PKC 2023, Part I. Lecture Notes in Computer Science, vol. 13940, pp. 177–204. Springer (2023). https://doi.org/10.1007/978-3-031-31368-4_7
7. Ducas, L., Postlethwaite, E.W., Pulles, L.N., van Woerden, W.: HAWK: Module LIP makes lattice signatures fast, compact and simple. In: Advances in Cryptology – ASIACRYPT 2022, Part IV. Lecture Notes in Computer Science, vol. 13794, pp. 65–94. Springer (2022). https://doi.org/10.1007/978-3-031-22972-5_3
8. Ducas, L., van Woerden, W.: On the lattice isomorphism problem, quadratic forms, remarkable lattices, and cryptography. In: Advances in Cryptology – EUROCRYPT 2022, Part III. Lecture Notes in Computer Science, vol. 13277, pp. 643–673. Springer (2022). https://doi.org/10.1007/978-3-031-07082-2_23
9. Espitau, T., Wallet, A., Yu, Y.: On Gaussian sampling, smoothing parameter and application to signatures. In: Advances in Cryptology – ASIACRYPT 2023, Part VII. Lecture Notes in Computer Science, vol. 14444, pp. 65–97. Springer (2023). https://doi.org/10.1007/978-981-99-8739-9_3
10. van Gent, D.M.H., Pulles, L.N.: HAWK: Having automorphisms weakens key. IACR Communications in Cryptology **2**(2) (2025). https://doi.org/10.62056/a3qjp2w9p
11. van Gent, D.M.H., van Woerden, W.: A search to distinguish reduction for the isomorphism problem on direct sum lattices. In: Advances in Cryptology – ASIACRYPT 2025. Lecture Notes in Computer Science, vol. 16247, pp. 3–33. Springer (2025). https://doi.org/10.1007/978-981-95-5099-9_1

12. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC). pp. 197–206. ACM (2008). https://doi.org/10.1145/1374376.1374407
13. Mureau, G.: Special genera of Hermitian lattices and applications to HAWK. In: Theory of Cryptography – TCC 2025. pp. 289–317. Lecture Notes in Computer Science, Springer (2025). https://doi.org/10.1007/978-3-032-12290-2_10
14. Mureau, G., Pellet-Mary, A., Pliatsok, G., Wallet, A.: Cryptanalysis of rank-2 module-LIP in totally real number fields. In: Advances in Cryptology – EUROCRYPT 2024, Part VII. Lecture Notes in Computer Science, vol. 14657, pp. 226–255. Springer (2024). https://doi.org/10.1007/978-3-031-58754-2_9
15. National Institute of Standards and Technology: Module-lattice-based digital signature standard. Federal Information Processing Standards Publication (FIPS) 204 (2024). https://doi.org/10.6028/NIST.FIPS.204
16. van Poppelen, A.: Cryptographic decoding of the Leech lattice. Cryptology ePrint Archive, Paper 2016/1050 (2016), master's thesis, Utrecht University. https://eprint.iacr.org/2016/1050
17. Prest, T.: Sharper bounds in lattice-based cryptography using the Rényi divergence. In: Advances in Cryptology – ASIACRYPT 2017, Part I. Lecture Notes in Computer Science, vol. 10624, pp. 347–374. Springer (2017). https://doi.org/10.1007/978-3-319-70694-8_13
18. Prest, T., Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z.: FALCON: Fast-fourier lattice-based compact signatures over NTRU. NIST Post-Quantum Cryptography Standardization (2020), https://falcon-sign.info
19. Silva, F.C., Lie, M., Ling, C.: On hull attacks on the module lattice isomorphism problem. Cryptology ePrint Archive, Paper 2025/1376 (2025), https://eprint.iacr.org/2025/1376
20. Straznickas, Z., Weis, S.A.: HAWK-n key recovery reduces to SVP in dimension $n/2+1$. Anthropic (Jul 2026), https://www.anthropic.com/document/hawk_key_recovery.pdf
21. Viazovska, M.S.: The sphere packing problem in dimension 8. Annals of Mathematics **185**(3), 991–1015 (2017). https://doi.org/10.4007/annals.2017.185.3.7

A Implementation Details

A.1 Bit Packed $\mathbb{F}_2$ Arithmetic

Coefficients are packed into 64 bit words, and each key precomputes a 4 bit comb table for f, g, F, G, reducing each negacyclic $\mathbb{F}_2$ product to a window scan over the changing challenge operand; negacyclic wraparound is an XOR into the low coefficients since $-1 = 1$ over $\mathbb{F}_2$. The public byte per coefficient layout is preserved at the interface.

A.2 Shared NTT Unit

The negacyclic transform matches the $X^n + 1$ reduction with a $2n$-th root of unity; pointwise products run in the Montgomery domain to avoid per multiply divisions. One unit serves both signer and verifier, which keeps the twiddle tables and Montgomery constants in cache. Exact two and three prime CRT helpers are available for reference cross checks.

A.3 One Prime Reconstruction

A deterministic bound, derived from the default verification threshold, the row norms of P_n^{-1} and the `int8` basis coefficient bound, places every reconstructed coefficient inside the centered range of the prime. A guarded hard limit $|w| <$ 140,000,000 rejects and restarts in the impossible case. This removes multi prime CRT from the hot path, where larger custom verification thresholds would need a new range proof or explicit rejection.

A.4 Rejection Ordering

Because acceptance is decided by $\|x_E\|^2$ alone, the norm test runs before reconstruction. Rejected samples are discarded before any reconstruction work, which both saves the discarded work and guarantees that the one prime range proof applies to every reconstructed output.

A.5 Prepared State and Concurrency

Prepared structures carry magic and $\log n$ fields, and debug builds bind them to (f, g, F, G) with a 64 bit digest; an unprepared entry point is also available. Worker randomness comes from one bound SHAKE stream per worker (section 6.1). The sampler reads only constant tables, but full signing calls are not reentrant across application threads, since the worker pool and its configuration are process global.

B The Galois Attack: Proofs and Experiments

This appendix proves proposition 4, apart from item (2), which section 5.4 proves, and lemma 8, and it reports the experiments of section 5.4. The notation is that of section 5.4.

B.1 Proof of proposition 4

By lemma 7(2) it suffices to treat $\Lambda(S_n)$.

Change of frame. Since $\tau(P_n) = P_n$, $\sigma(P_n)^T = P_n^\star$ and $\mathrm{adj}(P_n W P_n^{-1}) = P_n(\mathrm{adj}\, W)P_n^{-1}$, the map $W \mapsto P_n W P_n^{-1}$ preserves the first constraint and det, and turns $S_n W = \sigma(W)^T S_n$ into the second constraint for $Q = I$. The rational solutions of both constraints for $Q = I$ are the matrices $W(a,b) = \left(\begin{smallmatrix} a & b \\ \sigma(b) & \tau(a) \end{smallmatrix}\right)$ with $\sigma(a) = a$ and $\tau(b) = -b$, and $Q^{(\tau)}(W(a,b)) = \frac{1}{4}\,\mathrm{Tr}(a^\star a + b^\star b)$; the computation in [20, Sect. 4.3, Prop. 4.5] does not use integrality. So $\Lambda(S_n)$ is isometric to the lattice of those $W(a,b)$ with $P_n^{-1} W(a,b) P_n \in M_2(R_n)$, that is, with $W(a,b)M_n \subseteq M_n$.

Blocks. Let $k = n/4$ and write $w = \sum_{r<k} X^r w_r$ with $w_r \in K_8 = \mathbb{Q}(Y)$, entrywise for matrices. As $R_n = \bigoplus_r X^r O_8$ and P_n has entries in O_8 (lemma 3), $WM_n \subseteq M_n$ if and only if every W_r lies in $\mathfrak{O} = PM_2(O_8)P^{-1}$. Moreover $\tau(X^r w_r) = (-1)^r X^r w_r$, $(X^r w_r)^\star = Y^{-1} X^{k-r} w_r^\star$ for $0 < r < k$, and $\mathrm{Tr}(x^\star y) = k \sum_r \mathrm{Tr}_{K_8/\mathbb{Q}}(x_r^\star y_r)$ because $\mathrm{Tr}_{K_n/K_8}(X^j) = 0$ for $0 < |j| < k$; so the even and the odd components are orthogonal.

Even components. For even r, $b_r = \sigma(b)_r = 0$ and $\tau(a)_r = a_r$, so $W_r = a_r I$, which lies in $\mathfrak{O}$ if and only if $a_r \in O_8$. The even part is thus $\{W(a,0) : a \in R_n^\sigma \cap \mathbb{Z}[X^2]\}$, as for HAWK, with the orthogonal basis I and $W(X^{2j} - X^{n-2j}, 0)$, $0 < j < n/4$, of norms $n/4$ and $n/2$ [20, Prop. 4.5]: it is $\sqrt{n/4}\,(\mathbb{Z} \oplus \sqrt{2}\,\mathbb{Z}^{n/4-1})$.

Odd components. Since $4 \mid k$, the odd r form $n/16$ pairs $\{r, k-r\}$ with $r \neq k-r$. The conditions $\sigma(a) = a$ and $\sigma(b) = -b^\star$ give $W_{k-r} = -Y^{-1} W_r^\star$, with $W_r \in \mathfrak{sl}_2(K_8)$ free. Now $P^\star P = 4S'$ with $S' = \left(\begin{smallmatrix} 2+Y-Y^3 & Y^2 \\ -Y^2 & 2 \end{smallmatrix}\right)$ by eq. (11), and $\det S' = 3 + 2(Y - Y^3) = (1 + Y - Y^3)^2$ is a unit, so $S' \in GL_2(O_8)$. As $P^{-1} W^\star P = (P^\star W P^{-\star})^\star$ and $P^\star W P^{-\star} = S'(P^{-1} W P) S'^{-1}$, the order $\mathfrak{O}$ is closed under $\star$, and the pair contributes exactly the $W_r \in \mathfrak{sl}_2(K_8) \cap \mathfrak{O} = P\,\mathfrak{sl}_2(O_8)P^{-1}$. For $W_r = \left(\begin{smallmatrix} \alpha & \beta \\ \gamma & -\alpha \end{smallmatrix}\right)$ its contribution to $Q^{(\tau)}$ is $\frac{k}{4}\,\mathrm{Tr}_{K_8/\mathbb{Q}}(2\alpha^\star\alpha + \beta^\star\beta + \gamma^\star\gamma) = \frac{n}{4}\|W_r\|^2$, which proves eq. (37).

(1) Conjugation by P is a K_8-linear map of $\mathfrak{sl}_2(K_8)$ of determinant 1 (if P has eigenvalues λ_1, λ_2, those of the map are 1 and λ_1/λ_2, λ_2/λ_1), so L_{12} has the determinant of $\mathfrak{sl}_2(O_8) \cong \mathbb{Z}^8 \oplus \sqrt{2}\,\mathbb{Z}^4$, which is 16. The summands of eq. (37) then give $(n/4)^n\, 2^{n/4-1}\, 16^{n/16} = (n/4)^n 2^{n/2-1}$, as in [20, Prop. 4.5].

(3) We computed the Gram matrix of L_{12} in the basis PZP^{-1}, with Z running through $Y^j\left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$, $Y^j\left(\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right)$, $Y^j\left(\begin{smallmatrix} 0 & 0 \\ 1 & 0 \end{smallmatrix}\right)$, $0 \le j < 4$, in exact arithmetic. It is integral with determinant 16 and has exactly 120 vectors of norm 2, which split into two mutually orthogonal irreducible root systems of rank 6 with 60 roots each. By the ADE classification both are D_6, and $\det(D_6 \oplus D_6) = 16 = \det L_{12}$, so the root sublattice is all of L_{12}. □

B.2 Proof of lemma 8

(1) As in [20, Prop. D.2], $\tau(x) = V_\tau^{-1} x$ if and only if $\tau(Bx) = Bx$, and $\tau(x) = -V_\tau^{-1} x$ if and only if $\tau(Bx) = -Bx$; as $BR_n^2 = R_n^2$, the solutions are $B^{-1}O'^2$ and $B^{-1}(XO')^2$. (2) For $x = B^{-1}u$ and $x' = B^{-1}u'$, $T(x, x') =$

$\mathrm{Tr}(u^\star S_n u') = n\langle P_n u, P_n u'\rangle$. Under $X^2 \mapsto X$, O' becomes $R_{n/2}$ with coefficient norms preserved, and P_n becomes $P_{n/2}$, since $Y = X^{n/4} = (X^2)^{n/8}$; so $(M_{V_\tau}, T) \cong \sqrt{n}\, M_{n/2} \cong \sqrt{n}\,(2E_8)^{\oplus n/8}$ by lemma 3 in degree $n/2$, whose proof only needs $n/2 \geq 4$. Multiplication by X is an isometry, which gives the case $-V_\tau$. (3) By (1) the columns Bx, By of A lie in O'^2, $\det A = \det U = 1$, and $U^\star Q_{E_8} U = A^\star S_n A$; the entries of S_n lie in $\mathbb{Z}[Y] \subset O'$ and become those of $S_{n/2}$. □

B.3 Experiments

We checked the statements of section 5.4 at small degree in exact integer arithmetic (Python; no lattice library is used, with LLL at $\delta = 0.99$ and Fincke–Pohst enumeration). Keys follow HawkKeyGen's structure, with f, g centred binomial and F, G from an exact NTRUSolve. From the public key alone we computed a basis of $\Lambda(Q_{E_8})$ as an integer kernel and its Gram matrix under $Q^{(\tau)}$. For five keys each at $n = 8$, 16 and 32 and two at $n = 64$, the determinant is $(n/4)^n 2^{n/2-1}$; there is one pair of vectors of norm $n/4$, which is $\pm V_\tau$, and $8n-2$ vectors of norm $n/2$, whose root systems are $(n/4-1)A_1$ and $(n/8)D_6$. Together with $\pm V_\tau$ they span a sublattice of the same determinant, which certifies eq. (37). For one key per degree, the HAWK form $B^\star B$ of the same basis gives the expected $n/2+1$ pairs of norm $n/4$. LLL alone returned $\pm V_\tau$ as its first vector for 14 of these 17 keys, and enumeration below norm $n/4$ found it for all. For $n \geq 16$ the descent lattice M_{V_τ} has determinant $(4n)^n$, and at $n \leq 32$, scaled by $1/(4n)$, its norm 2 vectors form $n/8$ orthogonal root systems of 240 roots each, so it is $E_8^{\oplus n/8}$ as lemma 8 states. Finally, the τ-tower as described, with the Bézout step solved as an integer linear system, the new basis size reduced under the Hermitian form of Q_{E_8} as in [20, App. D], and no reduction of any descent lattice, recovered a B' with $B'^\star S_n B' = Q_{E_8}$ for every key, in under a second at $n \leq 32$ and in about 75 seconds at $n = 64$, nearly all spent on the top cocycle lattice; the wrong sign failed one level down each time. The script is part of the artifact `https://github.com/Maximilian-Adam/hawk_e8`. HAWK's parameter sets are far beyond such computations, and the experiments test the structure, not the cost.

C Further Measurements

The tables below complete section 8. Table 12 lists every object size, and table 13 is implementation context only. Table 14 is a separate run set on both CCDs, fair within itself but under different conditions from the headline runs, and never combined with them. Table 15 gives the cycles behind table 8.

Table 12. Sizes in bytes. HAWK's signature is compressed (s_1 only), E8-CM's uncompressed (salt, s_0 and s_1). Prepared private key: HAWK's decoded key, E8-CM's per-key NTT and $\mathbb{F}_2$ basis tables in a fixed 40,960-byte layout; decoded public key: q_{00} and q_{01} (E8-CM as 32-bit integers). `tmp`: the caller-supplied scratch of each call (E8-CM's functions also use stack buffers, not counted); a dash: the object does not exist for the scheme.

	$n = 256$		$n = 512$		$n = 1024$	
object	HAWK	E8-CM	HAWK	E8-CM	HAWK	E8-CM
private key, encoded	96	544	184	1,080	360	2,152
private key, prepared	592	40,960	1,184	40,960	2,368	40,960
public key, encoded	450	2,048	1,024	4,096	2,440	8,192
public key, decoded	784	2,064	1,568	4,128	3,136	8,256
signature, encoded	249	1,038	555	2,072	1,221	4,136
signature, decoded	526	–	1,048	–	2,088	–
`tmp`, key generation	5,639	6,151	11,271	12,295	22,535	24,583
`tmp`, signing	1,543	40,960	3,079	40,960	6,151	40,960
`tmp`, verification	2,567	2,055	5,127	4,103	10,247	8,199
`tmp`, fast verification	3,911	–	7,751	–	15,495	–

Table 13. Implementation context, not a fair pair: upstream HAWK's AVX2 build (AVX2) against its portable `ref` (HAWK) and E8-CM (`ref` signer, fast verifier), which has no AVX2 path. Thousands of core cycles on one core, same runs as tables 4 and 11; ratio E8-CM/AVX2, every 95% CI half-width at most 0.067. Key generation in table 10.

		`O2` (kcycles)				`O3native` (kcycles)			
n	operation	HAWK	AVX2	E8-CM	ratio	HAWK	AVX2	E8-CM	ratio
256	sign all-in	123.0	38.1	307.5	8.07	110.0	29.2	201.0	6.89
	sign prep.	126.8	46.0	253.2	5.51	112.8	37.8	159.5	4.22
	verify all-in	115.7	40.2	158.2	3.93	87.1	39.5	119.1	3.01
	verify prep.	105.9	33.5	135.4	4.05	78.3	32.7	98.5	3.01
512	sign all-in	257.5	68.3	626.1	9.16	219.7	51.3	401.2	7.82
	sign prep.	262.3	84.1	511.3	6.08	223.3	68.7	316.5	4.61
	verify all-in	247.4	78.6	331.7	4.22	175.0	76.8	234.9	3.06
	verify prep.	226.2	64.6	287.5	4.45	157.0	62.7	194.8	3.11
1024	sign all-in	565.3	144.2	1,319.2	9.15	468.8	107.9	836.8	7.76
	sign prep.	539.1	161.2	1,051.1	6.52	445.2	130.2	644.6	4.95
	verify all-in	530.8	161.6	697.8	4.32	356.3	157.7	473.2	3.00
	verify prep.	485.0	131.1	610.8	4.66	312.5	126.9	394.0	3.11

Table 14. Parallel signing latency across both CCDs, E8-CM/HAWK: a separate run set (5 runs per flag set) on all 16 cores, the V-cache CCD's 8 first, so $T \leq 8$ runs there (wall-clock cycles within 0.21% of the headline runs) and $T = 16$ spans both CCDs (split L3, V-cache on one CCD only). Fair within the run set, and never combined with the headline tables. Known asymmetry: E8-CM's sampler workers write $8n$ bytes per call, HAWK's $2n$, so at $T = 16$ E8-CM moves more data between the CCDs (an implementation property, not the lattice's). Columns as in table 8, CPU at $T = 16$; every 95% CI half-width is at most 0.012.

		`O2`				`O3native`			
n	acct	T=1	8	16	CPU	T=1	8	16	CPU
`ref-T`/`ref`									
256	all-in	2.47	2.05	1.99	2.09	1.67	1.87	1.86	1.52
	prep.	1.98	1.28	1.22	1.81	1.30	1.18	1.19	1.30
512	all-in	2.36	2.04	2.00	2.22	1.67	1.93	1.96	1.66
	prep.	1.90	1.26	1.20	1.87	1.30	1.19	1.20	1.38
1024	all-in	2.30	1.98	1.94	2.23	1.67	1.89	1.91	1.65
	prep.	1.90	1.28	1.22	1.87	1.33	1.23	1.22	1.35
`hot64`									
256	all-in	1.61	1.86	1.93	1.71	1.51	1.82	1.86	1.54
	prep.	1.14	1.09	1.17	1.39	1.11	1.12	1.18	1.30
512	all-in	1.59	1.86	1.92	1.54	1.50	1.86	1.92	1.44
	prep.	1.13	1.07	1.12	1.17	1.11	1.12	1.16	1.14
1024	all-in	1.58	1.83	1.86	1.52	1.50	1.84	1.88	1.42
	prep.	1.15	1.12	1.13	1.14	1.13	1.18	1.19	1.11
`hot32`									
256	all-in	1.72	1.89	1.95	1.84	1.61	1.86	1.89	1.72
	prep.	1.15	1.10	1.17	1.49	1.12	1.13	1.20	1.45
512	all-in	1.70	1.89	1.94	1.63	1.60	1.91	1.94	1.52
	prep.	1.13	1.08	1.12	1.20	1.11	1.13	1.17	1.17
1024	all-in	1.68	1.85	1.88	1.62	1.59	1.88	1.90	1.51
	prep.	1.16	1.12	1.13	1.16	1.14	1.18	1.19	1.13

Table 15. Thousands of wall-clock (TSC) cycles per signature with T threads, behind table 8. At $T = 1$ the TSC and core-cycle counts of every configuration agree within 1.4%.

			`O2` (kcycles)				`O3native` (kcycles)			
n	acct	scheme	T=1	2	4	8	T=1	2	4	8
`ref-T/ref`										
256	all-in	HAWK	124.4	95.4	80.7	73.8	120.0	87.1	70.7	62.7
		E8-CM	306.5	217.7	173.3	151.1	200.1	153.0	129.4	117.5
	prep.	HAWK	127.3	98.3	83.6	76.7	122.0	89.0	72.6	64.6
		E8-CM	252.2	164.2	120.0	97.6	158.8	111.8	88.2	76.2
512	all-in	HAWK	264.3	200.3	167.1	152.2	238.8	170.2	135.1	119.1
		E8-CM	624.0	443.4	353.1	310.0	399.7	301.9	252.4	229.6
	prep.	HAWK	268.3	204.2	170.9	156.1	241.5	172.9	137.6	121.7
		E8-CM	509.5	329.1	238.9	195.6	315.3	217.8	168.4	145.2
1024	all-in	HAWK	572.3	441.6	374.9	343.0	500.5	362.4	291.1	256.9
		E8-CM	1,316.1	950.8	768.1	679.5	834.3	632.9	532.6	484.1
	prep.	HAWK	550.4	419.5	352.7	320.8	481.0	343.0	271.4	237.3
		E8-CM	1,048.4	683.5	501.0	412.1	642.5	440.5	340.3	291.7
`hot64`										
256	all-in	HAWK	123.2	94.9	81.0	74.2	110.6	82.6	69.0	62.2
		E8-CM	198.0	162.5	144.9	137.6	166.4	135.1	119.8	113.3
	prep.	HAWK	126.1	97.8	83.9	77.1	112.5	84.5	70.9	64.1
		E8-CM	143.9	109.1	91.6	84.2	125.1	93.9	78.6	71.9
512	all-in	HAWK	257.2	196.9	165.9	152.3	222.2	162.1	131.5	118.1
		E8-CM	408.5	335.5	298.7	282.1	333.3	267.7	234.8	220.2
	prep.	HAWK	261.2	200.7	169.7	156.1	224.9	164.8	134.1	120.8
		E8-CM	293.9	221.0	184.4	167.6	249.0	183.6	150.8	135.8
1024	all-in	HAWK	557.8	434.2	372.0	342.1	466.6	344.5	282.8	253.0
		E8-CM	884.2	735.0	660.3	625.1	698.8	565.5	498.5	466.6
	prep.	HAWK	535.9	412.2	349.9	320.0	447.2	325.0	263.2	233.4
		E8-CM	616.6	467.6	393.3	357.6	507.5	374.1	307.1	274.6
`hot32`										
256	all-in	HAWK	100.8	84.5	75.3	71.5	88.8	72.6	63.4	59.5
		E8-CM	173.4	150.8	139.2	134.7	143.2	124.0	114.2	110.6
	prep.	HAWK	103.8	87.4	78.1	74.3	90.7	74.5	65.2	61.3
		E8-CM	119.6	97.5	85.9	81.4	101.8	82.8	73.0	69.2
512	all-in	HAWK	212.3	174.0	155.4	146.5	179.1	140.3	121.6	112.5
		E8-CM	359.7	311.4	287.2	276.6	286.8	244.3	223.8	214.6
	prep.	HAWK	216.3	177.9	159.2	150.3	181.8	143.0	124.1	115.1
		E8-CM	245.2	196.9	172.8	161.7	202.5	160.3	139.8	130.2
1024	all-in	HAWK	468.9	389.6	349.3	331.6	381.6	301.5	261.0	243.1
		E8-CM	786.7	686.7	636.4	613.7	605.5	518.8	475.3	455.7
	prep.	HAWK	446.9	367.6	327.2	309.5	362.1	282.0	241.5	223.4
		E8-CM	519.2	419.4	369.4	346.2	414.2	327.3	283.8	263.7