HAWK "Guessing Game" is not Polynomial-Time

Markku-Juhani O. Saarinen

Tampere University, Finland
markku-juhani.saarinen@tuni.fi

Abstract. We show that the runtime complexity of the attack described in *"Cryptanalysis of HAWK: a Guessing Game"* is much higher than originally claimed by its authors, and the attack is unlikely to pose a threat to HAWK's security in its present form. The attack algorithm had not been implemented before this work; the polynomial-time running-time claim was based on four "plausible heuristics". Our experiments and implementation data point to a super-polynomial class-number obstruction, consistent with exponential-scale growth. The experiments also helped to identify faulty "Heuristic 4" as the source of the observed computational wall when scaling dimension n. The authors of Guessing Game have acknowledged our findings. To make the argument more universal, we also offer a machine-checked conditional reduction from explicit assumptions that shows the complexity to be at least super-polynomial. In terms of methodology, our work demonstrates the role of powerful AI tools in contemporary cryptanalysis – the sudden feasibility of rapid exploration and trial implementation of advanced attack techniques. A public research artifact contains all source code and datasets to reproduce our results.

1 Introduction: HAWK and the Guessing Game

HAWK [2,4] is an efficient, quantum-resistant digital signature algorithm whose design and security analysis is closely connected to the Lattice Isomorphism Problem (LIP). In May 2026, HAWK was selected to continue in the current NIST standardization process for additional PQC signature schemes [1], and hence its security is under especially intense scrutiny.

Guessing Game. On June 26, 2026, a claimed probabilistic polynomial-time attack on HAWK was published as a preprint [6]: *"Cryptanalysis of HAWK: a Guessing Game"*. Despite the title claiming "cryptanalysis", in the text its authors did not fully claim to have broken HAWK, as they had not implemented their attack – it merely rested on "four plausible heuristics."

On the day of its publication, the author decided to try to implement the attack. This initial step primarily involved using the Anthropic Opus 4.8 and OpenAI GPT-5.5 models to write SageMath scripts that model the GuessingGame algorithm. We proceeded in small steps, cross-checking findings with unit tests

and other sanity checks. The original plan was to eventually move to a truly optimized implementation once the basic attack mechanism had been validated with toy parameter sets.

Experiments hit the wall. Over the course of a single weekend, we implemented and modeled most of the algorithm's steps in SageMath. However, we soon hit a wall running the algorithm as it became apparent that some of the heuristic assumptions of GuessingGame did not hold. After continued trials, the exponential runtime wall was triaged to "Heuristic 4". On June 28, 2026, we communicated our initial findings (largely the same text as in Section 3) to the authors of [6], who later updated the ePrint with the following note:

> *"**Update (30/06)**: Following discussions with Daniel Apon and Markku-Juhani Saarinen, we acknowledge that Heuristic 4 is insufficient to conclude that the main algorithm runs in polynomial time, and in fact the main algorithm appears to run in super-polynomial time. This mistake originates from the count of ideals of norm q' in $\mathcal{O}_F$: one must include fractional ideals in this count, of which there are many. We note as an aside that Heuristics 1–3 have been independently experimentally verified."*

Structure and contributions of this paper. We describe the GuessingGame attack and its claimed heuristics in Section 2. Section 3 provides heuristic analysis and evidence pointing to a super-polynomial class-number obstruction, consistent with exponential-scale growth. To make the argument more universal, Section 4 contains a simple machine-checked conditional reduction to super-polynomial complexity of the GuessingGame algorithm. Section 5 discusses our reproducibility artifacts, toy key-recovery PoC, and the machine-checked proof structure, followed by Conclusions (including a commentary on the impact of AI on cryptanalytic work) in Section 6.

2 The Structure and Heuristics of Guessing Game

The GuessingGame algorithm [6, Algorithm 2] is reproduced here as Algorithm 1. We observe that GuessingGame has a randomized loop-until-succeed structure with early aborts ("filtering"). The algorithm uses some fairly advanced tools: The reduction of HAWK to the principal ideal problem in a quaternion algebra from EUROCRYPT 2025 [3] on Line 12, as well as CM-order principal-ideal computation with the Lenstra-Silverberg algorithm [5] on Line 25. Both of these subroutines run in classical probabilistic polynomial time (PPT). Hence, the overall complexity of GuessingGame is polynomial if the number of iterations for reaching a "win" is polynomially bounded.

In their own analysis of the algorithm, the authors of GuessingGame identify four number-theoretic heuristics on which its success in PPT depends. These were framed as open questions. We reproduce Heuristics 1–4 from [6, Section 1.2] here as our discussion is largely centered on them:

Algorithm 1: GuessingGame (Reproduced from [6, Algorithm 2]).

Require: A valid HAWK public key $G \in M_2(\mathcal{O}_L)$ with corresponding pseudo-Gram matrix $\mathcal{G} := (G, \mathcal{O}_L, \mathcal{O}_L)$ and some $\sigma \approx 2$.
Ensure: $C \in M_2(\mathcal{O}_L)$ with $C^*C = G$.

win $\leftarrow 0$
while win $= 0$ **do**
  $x_r \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}$ for $1 \le r \le n/2$
  $x \leftarrow \sum_{r=1}^{n/2} x_r \zeta_n^r$
  $U \leftarrow \left(\begin{smallmatrix} 1 & 0 \\ x & 1 \end{smallmatrix}\right)$
  $G' \leftarrow U^* G U$ ▷ 'Close' rank-2 module-LIP instance
  $q' \leftarrow G'_{11}$
  if $\|q'\|_2 > \Omega(n^{11/2})$ **then**
    continue ▷ [6, Sec. 6]
  end
  $\mathbf{G}' \leftarrow (G', \mathcal{O}_L, \mathcal{O}_L)$
  $\mathcal{I} \leftarrow \Phi(\mathbf{G}')$ ▷ Form the $\mathcal{O}$-nrdPIP instance
  if $q'\mathcal{O}_K$ *is not a prime ideal* **then**
    continue
  end
  Compute $\beta_0 \in \mathcal{O}$ with $\mathcal{I} = (\beta_0, q')\mathcal{O}$ ▷ [6, Sec. 5]
  $\mathcal{B} \leftarrow \{q'\zeta_n^r,\ q'j\zeta_n^r : 1 \le r \le n/2\}$
  Reduce β_0 using $\mathcal{B}$ to obtain β with $\mathcal{I} = (\beta, q')\mathcal{O}$, $\|\beta\|_2 < O(n^{13/2})$ ▷ Sec. 6
  $\alpha \leftarrow -\operatorname{nrd}(\beta) + \operatorname{trd}(\beta)^2/4$
  $F \leftarrow K(\sqrt{\alpha})$ ▷ Enter the number field setting
  if $\mathcal{O}_F \ne \mathcal{O}_K[\sqrt{\alpha}]$ **then**
    continue ▷ [6, Sec. 7]
  end
  Let $\chi : K(\beta) \to K(\sqrt{\alpha})$ be an explicit morphism of fields.
  Apply the Lenstra–Silverberg algorithm ([6, Theorem 5]) to $A = \mathcal{O}_F$, $a = (\operatorname{trd}(\beta)/2 + \sqrt{\alpha}, q')\mathcal{O}_F$, and $w = q'$, returning $h \in F$ with $\chi^{-1}(h) \in \mathcal{I}$ and $N_{F/K}(h) = q'$, assuming that such an element exists. In particular, since $\operatorname{nrd}(\chi^{-1}(h)) = N_{F/K}(h) = q'$, we have $\mathcal{I} = \chi^{-1}(h)\mathcal{O}$ ▷ [6, Sec. 8]
  if $h = \perp$ **then**
    continue ▷ Check nrdPIP instance valid
  end
  $g \leftarrow \chi^{-1}(h)$
  Use the knowledge of a principal generator $\mathcal{I} = g\mathcal{O}$ to recover $C' := CU$ with $C'^*C' = G'$ ▷ [6, Sec. 10]
  win $\leftarrow 1$
end
return $C'U^{-1}$ ▷ Recover the HAWK private key

Heuristic H1 *Let α be sampled as in the main algorithm (Algorithm 1). The probability that α is square-free is lower bounded by $1/2$.*

Heuristic H2 *Let $K = \mathbb{Q}(\zeta_{2^l} + \zeta_{2^l}^{-1})$ for some $l \in \mathbb{N}$, and write $\mathfrak{d} \subseteq \mathcal{O}_K$ for the unique prime ideal of $\mathcal{O}_K$ lying over 2. There is a natural isomorphism $\mathcal{O}_K/\mathfrak{d}^2 \cong \mathbb{F}_2[X]/(X^2)$. The image of α in the quotient $\mathcal{O}_K/\mathfrak{d}^2$ corresponds to $X + 1 \in \mathbb{F}_2[X]/(X^2)$ with probability around $1/4$.*

Heuristic H3 (Restated) *Let $K = \mathbb{Q}(\zeta_{2^l} + \zeta_{2^l}^{-1})$ for some $l \in \mathbb{N}$, and let $S \subset \mathrm{Pl}(K)$ consist of all finite places dividing q', all places ramified in $K(\sqrt{\alpha})/K$, and all archimedean places of K. Write $U_{K,S}$ for the S-units in K. There is a map $\mathrm{Log} : U_{K,S} \to \mathbb{R}^{\#S}$, and one can construct a pair of lattices $L'' \subset L'$ in $\mathbb{R}^{\#S}$ such that, with overwhelming probability,*

$$|L' : L''| = O\Big(n^{2+o(1)}\Big).$$

The image of $\mathrm{Log}(q')$ in L'/L'' is 0 with probability $\Omega\big(n^{-2-o(1)}\big)$.

Heuristic H4 (Restated) *Let $K = \mathbb{Q}(\zeta_{2^l} + \zeta_{2^l}^{-1})$ and suppose that F is a field isomorphic to $K(\sqrt{\alpha})$. During the main algorithm (Algorithm 1), one forms an ideal $\mathfrak{a} \subseteq \mathcal{O}_F$ that depends on both q' and α, and $N_{F/K}(\mathfrak{a}) = q'\mathcal{O}_K$. In [6, Section 9] the authors show that, with overwhelming probability, there are $O\big(n^{1+o(1)}\big)$ ideals of $\mathcal{O}_F$ with relative norm $q'\mathcal{O}_K$. Given that one of these ideals is principal, the probability that $\mathfrak{a}$ itself is principal is $\Omega\big(n^{-1-o(1)}\big)$.*

3 Experimental Observations

The attack's per-guess success requires the specific ideal

$$\mathfrak{a} = \big(\mathrm{trd}(\beta)/2 + \sqrt{\alpha}, q'\big)\,\mathcal{O}_F \tag{1}$$

to be principal in $\mathcal{O}_F$. Modeling its ideal class as approximately uniform in Cl_F, this has probability about $1/h_F$, where h_F is the class number of the sampled CM field $F = K(\sqrt{\alpha})$. Equivalently, $h_F = h_{F/K}$ under Weber's conjecture, as assumed in the Guessing Game paper, and in all our tested cases. Combining the paper's claimed H3 and H4 rates with that class-group model would require, as a consequence of the model, an effective

$$h_F \le n^{3+o(1)}. \tag{2}$$

Our evidence – empirical at $n = 8, 16$, plus a relative class-number heuristic – is that h_F is super-polynomial, heuristically

$$h_F = 2^{\Theta(n \log n)} = n^{\Theta(n)}. \tag{3}$$

Thus the per-guess binding probability is about $1/h_F$, and GuessingGame does not run in the stated probabilistic polynomial time.

Empirically the deficit is localized in Heuristic 4, not Heuristic 3: H3 (q' is a relative norm) holds at or above its claimed rate, while the binding event (the specific $\mathfrak{a}$ is principal) essentially never fires once h_F is large.

3.1 The binding event

GuessingGame succeeds for a guess (U, β) only if the ideal it constructs,

$$\mathfrak{a} = \left(\mathrm{trd}(\beta)/2 + \sqrt{\alpha}, q'\right) \mathcal{O}_F, \tag{4}$$

is principal in $\mathcal{O}_F$ with a generator of relative norm q'. Equivalently, under the isomorphism $\chi : K(\beta) \to K(\sqrt{\alpha})$, the pullback $(\beta, q')\mathcal{O}_{K(\beta)}$ is principal and solves the O-nrdPIP. So the per-guess binding probability is

$$P_{\mathrm{bind}} = \Pr([\mathfrak{a}] = 1 \text{ in } \mathrm{Cl}_F) \approx \frac{1}{h_F}, \tag{5}$$

modeling $[\mathfrak{a}]$ as approximately uniform in Cl_F. We confirmed this model at reduced parameters, using toy keys and public $\Phi(G')$ built from public data only:

- **Positive control:** whenever $h_F = 1$, the specific $\mathfrak{a}$ is principal (7/7 such fields at $n = 8$). The test is correct and the mechanism is real.
- Per-$\mathfrak{a}$ principality is consistent with, and no larger than, the $1/h_F$ scale. In fact it is slightly below it: the candidate classes are mildly biased away from principal, which is conservative here.

3.2 What the heuristics require of h_F

The F-stage per-guess success factors as

$$P_{\mathrm{success}} = \Pr(\mathrm{H3}) \cdot \Pr(\mathrm{H4} \mid \mathrm{H3}), \tag{6}$$

with the paper's restated rates

$$\Pr(\mathrm{H3}) = \Omega\left(n^{-2-o(1)}\right), \qquad \Pr(\mathrm{H4} \mid \mathrm{H3}) = \Omega\left(n^{-1-o(1)}\right). \tag{7}$$

Thus the H3–H4 product in the paper estimates this F-stage rate at

$$P_{\mathrm{success}} = \Omega\left(n^{-3-o(1)}\right). \tag{8}$$

The event the attack actually needs is that the specific $\mathfrak{a}$ is principal. The paper's H3–H4 heuristics estimate its probability as that product, $\Omega(n^{-3-o(1)})$; the class-group model estimates the same event at approximately $1/h_F$. For the two estimates to agree – that is, for the heuristics to deliver the binding rate the attack needs – one needs

$$\text{effective } h_F \le n^{3+o(1)}. \tag{9}$$

This keeps H1/H2, H3, the H4 class event, and the exact-norm unit correction cleanly separated. The abstract's polynomial run time rests on this h_F being polynomially bounded.

3.3 Evidence that h_F is super-polynomial

Heuristic: relative class number of a CM extension. The extension F/K is CM, and F and K have equal unit rank, so the relative regulator contribution is an explicit $2^{O(n)}$ factor (the Hasse unit index $Q \in \{1, 2\}$ and powers of 2), not a source of $2^{\Theta(n \log n)}$ cancellation. In the relative class-number formula, the discriminant factor is roughly

$$\sqrt{\frac{|\Delta_F|}{|\Delta_K|}} = \sqrt{|\Delta_K| \cdot \mathrm{N}_{K/\mathbb{Q}}(\delta_{F/K})}, \tag{10}$$

since

$$\Delta_F = \Delta_K^2 \cdot \mathrm{N}_{K/\mathbb{Q}}(\delta_{F/K}). \tag{11}$$

Already $|\Delta_K| = 2^{\Theta(n \log n)}$ for the real cyclotomic subfield. Thus, absent extraordinary cancellation from the $L(1)$-terms, one expects

$$h_F = 2^{\Theta(n \log n)} = n^{\Theta(n)} \tag{12}$$

up to standard analytic caveats.

We deliberately do *not* invoke the abelian minus-class-number / Bernoulli product formula here: the sampled fields $F = K(\sqrt{\alpha})$ are generally non-abelian CM fields, so that formula does not apply.

Measurement: consistent, and already past n^3. Computing h_F over genuinely monogenic, H1/H2-passing instances gives:

n	fields	geom. mean h_F	n^3	Notes
8	~ 2500	242	512	Comparable; $n = 8$ alone looks consistent.
16	~ 50	87330	4096	About 20× past n^3; 48/51 fields exceed n^3.

The geometric mean, i.e. $\exp((1/m)\sum_i \log h_{F_i})$ over the sampled fields, jumps $242 \to 87330$ for one doubling of n. This already contradicts the effective n^3 scale at $n = 16$ in these experiments. Two reduced-size points cannot establish an asymptotic rate; we read this as the data being consistent with the relative class-number heuristic predicting $2^{\Theta(n \log n)}$, and inconsistent with the $O(n^3)$ scale the heuristics would need. The data alone cannot separate $2^{\Theta(n \log n)}$ from a high-degree polynomial, but either contradicts n^3.

3.4 The deficit is in Heuristic 4, not Heuristic 3

A 2×2 table at $n = 8$, prime q', over 238 monogenic survivors (H3 via Hilbert symbols at all places; H4 via `bnfisprincipal`, validated by idealnorm($\mathfrak{a}$) = $\mathrm{N}(q')$):

	H4=true	H4=false
H3=true	0	79
H3=false	0	159

So H3 holds with probability about 0.33, well above its claimed $\Omega(n^{-2})$, but the specific $\mathfrak{a}$ is principal in 0 of the 79 H3-true cases. H3 only requires q' to be a relative norm, meaning that some principal ideal of relative norm $q'\mathcal{O}_K$ exists somewhere. H4 requires the specific $\mathfrak{a}$ to be that principal ideal. Thus $\Pr(\text{H4} \mid \text{H3})$ appears to be governed by $1/h_F$, not n^{-1}; the conditional in Heuristic 4 is where the class-group obstruction enters.

We also tested enlarging the candidate count M: composite square-free q' gives $M = 2^{r-1}$ sign-patterns, but $r \lesssim (1 + o(1)) \log n$ by Erdos–Kac, so $M \lesssim \text{poly}(n)$ cannot offset a super-polynomial h_F, and per-pattern principality is if anything below the $1/h_F$ model.

4 Reduction to the class number

The measurements of Section 3 indicate that the H4 binding event occurs with probability $\approx 1/h_F$ in GuessingGame, and are consistent with the class-number heuristic that h_F grows faster than any fixed power of n. The algorithm itself is largely heuristic, and the particular distributions are non-trivial to analyze, so we tried to isolate assumptions sufficient to prove super-polynomial time complexity. This reduction is machine-checkable (Section 5.3).

Definition 1 (Asymptotic classes). *A function $f : \mathbb{N} \to \mathbb{R}_{>0}$ is* super-polynomial *if $n^k/f(n) \to 0$ for every $k \in \mathbb{N}$, and* polynomially bounded *if $f(n) = O(n^k)$ for some $k \in \mathbb{N}$.*

4.1 The binding event

Using the notation of Section 3, fix the dimension n and condition on the early filters of GuessingGame, up to a guess (U, β) that has produced

$$F = K(\sqrt{\alpha}), \qquad \mathfrak{a} = \big(\mathrm{trd}(\beta)/2 + \sqrt{\alpha},\, q'\big)\mathcal{O}_F, \qquad \mathrm{N}_{F/K}(\mathfrak{a}) = q'\mathcal{O}_K. \tag{13}$$

Let HE_4 be the event that $\mathfrak{a}$ passes H4 — that it is principal with a generator of relative norm q' — and put $p_{\mathsf{succ}}(n) = \Pr[\mathsf{HE}_4]$ under one guess with this conditioning. Every winning guess passes H4, so the one-guess win probability obeys $p_{\mathsf{win}}(n) \le p_{\mathsf{succ}}(n)$, and it suffices to bound p_{succ}.

Lemma 1 (Class event). HE_4 *implies* $[\mathfrak{a}] = 1$ *in* $\mathrm{Cl}(\mathcal{O}_F)$*, and hence*

$$p_{\mathsf{succ}}(n) \;\le\; \Pr\big[[\mathfrak{a}] = 1 \text{ in } \mathrm{Cl}(\mathcal{O}_F)\big]. \tag{14}$$

Proof. An integral ideal is trivial in the class group iff it is principal (mathlib's `ClassGroup.mk0_eq_one_iff`). Passing HE_4 requires $\mathfrak{a}$ principal, i.e. $[\mathfrak{a}] = 1$; the further exact-norm condition $v\bar{v} = q'$ enforced by the Lenstra–Silverberg step only shrinks the event.

Remark 1. Since the relative norm of $\mathfrak{a}$ is fixed to $[q'\mathcal{O}_K]$, the class $[\mathfrak{a}]$ ranges over the corresponding fiber of $\mathrm{N}_{F/K} : \mathrm{Cl}(\mathcal{O}_F) \to \mathrm{Cl}(\mathcal{O}_K)$, a torsor under the relative class group $\mathrm{Cl}_{F/K} = \ker \mathrm{N}_{F/K}$. Under Weber's conjecture $h_K = 1$, so $|\mathrm{Cl}_{F/K}| = h_F$ and the principal class is one point of a fiber of size h_F. Thus p_{succ} is controlled by a class-group density of order $1/h_F$ — not by the norm-existence density of H3, which asks only that q' be *some* relative norm.

4.2 The assumptions

Our reduction rests on a distributional assumption and a class-number growth assumption. The separate candidate-multiplicity input is elementary once the number of prime ideal factors of $q'\mathcal{O}_K$ is controlled.

Assumption 1 (Non-concentration) *A guess exposes at most $M(n)$ candidate ideals, and their classes do not concentrate on the principal class: the probability that some candidate is principal satisfies*

$$\Pr\big[[\mathfrak{a}] = 1\big] \;\le\; \frac{M(n)}{h_F(n)}. \tag{15}$$

Assumption 2 (Growth of the class number) *h_F is super-polynomial in n.*

Remark 2. Assumption 1 is conservative in the measurements: the observed classes sit if anything *below* the $1/h_F$ scale, biased away from principal, and the mechanism is confirmed by the positive control $h_F = 1 \Rightarrow \mathfrak{a}$ principal (Section 3). Assumption 2 is the relative class-number heuristic $h_F = 2^{\Theta(n \log n)}$, consistent with the measured $h_F \gg n^3$ at $n = 16$; for the weaker Proposition 1 below it is not needed at all.

4.3 The reduction

Lemma 2 (Candidate multiplicity). *Let $r(n)$ be the number of prime ideal factors of $q'\mathcal{O}_K$ exposed by a guess. If $r(n) = O(\log n)$, then the sign-pattern family $M(n) = 2^{r(n)-1}$ is polynomially bounded.*

Proof. If $r(n) \le C \log_2 n$ eventually, then $M(n) = 2^{r(n)-1} \le n^C$. The analysis of [6] gives $r(n) \le (1 + o(1)) \log n$ with overwhelming probability. In the Lean file we also formalize the cruder integer fact $2^{\omega(m)} \le m$; it is useful as a sanity check, but the prime-ideal-factor bound is the relevant justification for GuessingGame's actual q'.

Theorem 1 (Super-polynomial lower bound). *Assume Assumption 1, Assumption 2, and that $M(n)$ is polynomially bounded as in Lemma 2. Then $n^k\, p_{\mathsf{win}}(n) \to 0$ for every $k \in \mathbb{N}$; equivalently $1/p_{\mathsf{win}}$ is super-polynomial. Hence the expected number of guesses grows super-polynomially, and* GuessingGame *does not run in probabilistic polynomial time.*

Proof. Lemma 1 and Assumption 1 give $p_{\mathsf{win}}(n) \le p_{\mathsf{succ}}(n) \le M(n)/h_F(n)$, and Lemma 2 gives $M(n) \le C\, n^d$ for some C, d. Fix k. Then

$$0 \;\le\; n^k\, p_{\mathsf{win}}(n) \;\le\; \frac{C\, n^{k+d}}{h_F(n)} \;\longrightarrow\; 0, \tag{16}$$

the limit because h_F is super-polynomial (Assumption 2). As k was arbitrary, $1/p_{\mathsf{win}}$ is super-polynomial.

Proposition 1 (The clean n^{-3} scale already fails). *Assume Assumption 1 and that $M(n)\, n^3/h_F(n) \to 0$ — that is, that h_F/M asymptotically overtakes the n^3 scale. Then $p_{\mathsf{succ}}(n) = o(n^{-3})$, so the clean n^{-3} F-stage scale implied by the H3–H4 product does not hold.*

Proof. By Lemma 1 and Assumption 1, $0 \le n^3 p_{\mathsf{succ}}(n) \le M(n)\, n^3/h_F(n) \to 0$.

Remark 3. Proposition 1 is the firmer finite-scale warning: it invokes only Assumption 1 and the asymptotic condition $M(n)n^3/h_F(n) \to 0$. The measurements of Section 3 show that h_F is already above the n^3 scale at $n = 16$, but the proposition itself is still asymptotic. Theorem 1 is stronger — it excludes probabilistic polynomial time outright — but is conditional on Assumption 2, whose warrant is the class-number heuristic together with the measurements of Section 3, not a theorem.

5 Reproducibility

Artifacts are available from `https://github.com/mjosaarinen/gg-artifact`.

5.1 Experimental measurements (Sage/Python)

The directory `gg-sage/` contains the scripts and data used for Section 3. The shipped JSON records in `gg-sage/data/` are the exact records behind the class-number tables. They can be re-analysed without Sage:

```
cd gg-sage
python3 analyze_h4.py
```

This reproduces the reported geometric means $h_F \approx 242$ for $n = 8$ and $h_F \approx 87330$ for $n = 16$, and the small-class-number positive control $h_F = 1 \Rightarrow \mathfrak{a}$ principal in 7/7 cases. The Sage drivers in the same directory regenerate the data and cross-check the filtering gates; for example `run_h4_study.sage` regenerates the h_F records, `run_h3h4_xcheck.sage` compares the H3 and H4 gates on the same survivors, and `measure_heuristics.sage` and `measure_composite_q.sage` reproduce the auxiliary rate measurements. Regeneration is slower than re-analysis, especially for $n = 16$, because it recomputes class groups.

5.2 Key-recovery proof-of-concept

The subdirectory `gg-sage/gg-poc/` contains an end-to-end public-data recovery prototype at reduced parameters. Its purpose is a correctness witness for the algebraic recovery chain, not evidence of scalability: it succeeds only in the small-h_F positive-control regime. A representative run is

```
cd gg-sage
sage gg-poc/recover_pubport.sage n=4 seed=7 max_hf=8 \
    trials=200 qbits=22
python3 gg-poc/run_poc.py --recover
```

The recovery starts from the public Gram data, constructs the public ideal $\Phi(G')$, obtains a principal generator through the H4 principal-ideal step, pulls it back to the quaternion order, and outputs a private matrix C. The script verifies the public certificate $C^*C = G$, together with the ideal-side checks $g \in I$, $g\mathcal{O} = I$, and $\mathrm{nrd}(g) = q'$. The pinned fixtures are regression-tested by

```
python3 gg-poc/tests/test_recovery_wins.py
```

The default recovery path uses public data only. Driving the prototype from a fresh genuine HAWK reference key is optional and additionally requires the external `ref/hawk-py` implementation and its Python dependencies.

5.3 Reduction: Machine-checked formalization (Lean 4 / mathlib)

Lemmas 1 and 2, Theorem 1, and Proposition 1 have been formalized and machine-checked in Lean 4.31.0, mathlib `v4.31.0` (`gg-lean/H4Invalidity.lean`). The formalization can be checked with:

```
cd gg-lean && lake build H4Invalidity
```

Structure. Lemma 1 is `h4_class_event` (mathlib's `ClassGroup.mk0_eq_one_iff`); Lemma 2's polynomial candidate-multiplicity input is carried by the explicit hypothesis `hM : PolyBounded M`. The file also proves `two_pow_omega_le` and `candidateCount_polyBounded` as a crude sufficient-condition sanity check; Theorem 1 is `main` and Proposition 1 is `stated_rate_refuted`. The main assumptions are carried openly as hypotheses of `main`: Assumption 2 is `hHF`, and Assumption 1 is the non-concentration half of the reduction bound `hbound` — whose other half is the proved class event (Lemma 1) — while polynomial candidate multiplicity is the explicit hypothesis `hM`. The file also declares two unused placeholder axioms (`clsQuat`, `eichler_mass_superpoly`) for a separate quaternion-side statement; `#print axioms` on the displayed theorems does not depend on them.

Lean identifier	statement	status
`h4_class_event`	Lemma 1	proved
`two_pow_omega_le` *et al.*	crude multiplicity bound	proved
`main`	Theorem 1	proved from its hypotheses
`stated_rate_refuted`	Proposition 1	proved from its hypotheses
`hbound` (non-conc. half)	Assumption 1	*assumption*
`hHF : SuperPoly hF`	Assumption 2	*assumption*
`hM : PolyBounded M`	Lemma 2 input	*assumption*

Supplementary Lean file. The artifact also ships `gg-lean/H4Exponential.lean`, a supplementary exponential-strength conditional reduction. It is not used for the main super-polynomial theorem because it requires stronger open inputs, including an exponentially large relative class-group denominator and an exponentially small exceptional-field mass.

Not modelled. "Probabilistic polynomial time" concerns a randomized algorithm's expected running time, for which mathlib has no mature cost model; the checked statement is that the expected number of guesses $1/p_{\mathsf{win}}$ is super-polynomial. Multiplying by the polynomial per-iteration cost (the post-filter subroutines run in probabilistic polynomial time) recovers the running time, but that step is informal.

6 Conclusions

Based on evidence presented, we conclude that *"Cryptanalysis of HAWK: a Guessing Game"* [6] poses no direct threat to HAWK's security. To our knowledge, we were the first to implement key aspects of this attack and experimentally observe its behavior. We also offer a simple reduction that shows super-polynomial (infeasible) runtime under reasonable assumptions.

The implementation and analysis process made extensive use of AI tools. While a professional cryptographer and a trained mathematician, the author cannot claim to be a subject-matter expert in the CM-fields and quaternion-algebra techniques of [3] or in the Lenstra-Silverberg algorithm [5] required for the implementation. However, with appropriate planning and cross-validation techniques, one can use AI tools in conjunction with mathematical software such as SageMath (symbolic manipulation, algebraic objects, number-theoretic algorithms, etc) and Lean (formal proofs) to have reasonable confidence in the correctness of the implementation.

Given the technical complexity of the task, we cannot guarantee that our interpretation of the algorithms is flawless. It is worth noting that we started out with a good-faith attempt to make the attack work, and we are making our initial source code artifacts available for validation (Section 5). The authors of [6] have acknowledged the basic oversight in Heuristic 4 that we reported, which originally led them to overstate the attack's efficacy.

To the author, it is clear that this type of implementation project would not have been possible in such a short period before the current generation of Frontier

Models (namely Opus 4.8 and GPT-5.5), which have very high mathematical-analytical capabilities and fluent ability to use external tools such as SageMath. Heavily AI-assisted attack exploration is likely to be a central contributor to many cryptanalytic successes of the upcoming years.

References

1. Alagic, G., Bros, M., Ciadoux, P., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Liu, Y.K., Miller, C., Moody, D., 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. Interagency or Internal Report NIST IR 8610, National Institute of Standards and Technology (May 2026). https://doi.org/10.6028/NIST.IR.8610
2. 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 version 1.1 (february 5, 2025) (February 2025), https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-2/spec-files/hawk-spec-round2-web.pdf
3. Chevignard, C., Mureau, G., Espitau, T., Pellet-Mary, A., Pliatsok, H., Wallet, A.: A reduction from hawk to the principal ideal problem in a quaternion algebra. In: Fehr, S., Fouque, P. (eds.) Advances in Cryptology - EUROCRYPT 2025 - 44th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Madrid, Spain, May 4-8, 2025, Proceedings, Part II. Lecture Notes in Computer Science, vol. 15602, pp. 154–183. Springer (2025). https://doi.org/10.1007/978-3-031-91124-8_6, https://eprint.iacr.org/2025/287
4. Ducas, L., Postlethwaite, E.W., Pulles, L.N., van Woerden, W.P.J.: Hawk: Module LIP makes lattice signatures fast, compact and simple. In: Agrawal, S., Lin, D. (eds.) Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, 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, https://eprint.iacr.org/2022/1155
5. Lenstra Jr., H.W., Silverberg, A.: Testing isomorphism of lattices over cm-orders. SIAM J. Comput. **48**(4), 1300–1334 (2019). https://doi.org/10.1137/17M115390X, http://arxiv.org/abs/1706.07373
6. Nelson, B., Limbrey, J., Ling, C., Mendelsohn, A.: Cryptanalysis of HAWK: a guessing game. Cryptology ePrint Archive, Paper 2026/1318 (June 2026), https://eprint.iacr.org/2026/1318