SEMINAR · CWI · CRYPTOLOGY GROUP

Math Ability has Consequences

20261021-cwi-math-consequences

https://mjos.fi/doc/20261021-cwi-math-consequences.pdf

Markku-Juhani O. Saarinen
Tampere University, Finland
markku-juhani.saarinen@tuni.fi

TLP:CLEAR

CONTENTS

001 Introduction

002 HAWK: The Lee Sedol Moment

003 NGCC: China’s PQC & HASH algorithms

004 A Bad Year for McEliece

005 Quick Conclusions

SECTION

001

INTRODUCTION

ATTACKING ALGORITHMS, NOT SAMPLES / DATA

AI CRYPTANALYSIS -VS- ML CRYPTANALYSIS

ML Cryptanalysis trains a Deep Neural Network (DNN), Transformer, or other ML model directly with *plaintext / ciphertext / key material samples*, and then challenges the model with a key recovery task or a distinguishing task.

Examples: Gohr's 2019 SPECK attack [14], Salsa LWE attacks from 2022 [28].

DEFINITION (CASUAL CURRENT USE)

AI Cryptanalysis uses Large Language Models (LLMs) and related methods to analyze *cryptographic algorithms* to discover and to implement cryptanalytic attacks, methods, and related abstract theory.

Examples: Anthropic's 2026 HAWK attack [25], some results in this talk.

FRONTIER MODELS ARE REALLY GOOD NOW

Around mid-2025, I was struck by respected cybersecurity professionals admitting that frontier models are better at finding vulnerabilities than they are.

Many contributing factors for radical improvements in cryptographic abilities:

- Chain-of-Thought (intermediate reasoning), and tool use to do math.
- Math is ideal for Reinforcement Learning with Verifiable Rewards (RLVR).
- Mixture of Experts (MoE) or Model Routing, often with cryptography now.

  *Trained on, among other things, **all** IACR ePrints, arXiv, GitHub repositories, etc. – the models have extremely broad knowledge of math and crypto.*
- Quite obvious: Exponential spending on data centers & model training.

ADD ANY AMOUNT OF TESTING AND VERIFICATION

Base level – note how there are *two* implementations of every scheme:

1. Python “executable pseudocode” – a deterministic, unoptimized, readable model that maps directly to the paper. Used to create unit-level test vectors.
2. An optimized, native Rust implementation that may use machine intrinsics or Rayon multi-threading. Checked to reproduce the same test vectors.

Plus a lot of unit tests! I’d expect that even casual coding can now move to full formal models and logical equivalence checking – previously “too laborious”.

Hardware too: I recently vibe-coded a vector processor (Verilog/FPGA). A key enabler was the formal Sail model of RISC-V ISA. https://karucore.com/

LET'S USE AI TO RUN CRYPTANALYSIS EXPERIMENTS

```
read 2026-1318.pdf, check for correctness, and write a Sage prototype!
```

- In reality, you can't reliably "one-shot" anything very advanced. It's an interactive process where you drive the activities; ask for code, tests, etc.
- I'm a terminal guy and use CLI coding harnesses almost exclusively: Claude Code, Codex, Antigravity, OpenCode – the CLIs are very similar.
- A sandboxed environment with the necessary tools for the models to use; Full SageMath, Rust toolchain, Lean, etc + a big stash of research PDFs.
- I've found it **extremely** useful to have different (vendor) models cross-check each other. Just like in "pair programming" – they have different training.

WHAT IS ALL OF THIS TALK OF “LEAN” ?

```
open Complexity OpenAI -- NP-hard N^c-GapCVP for fixed c < 1/2
theorem hardnessBelowHalf (c : Real) (hc : 0 < c)
   (h : c < (1 : Real) / 2) :
   GapCVP.NPHardPromise (integerTargetGapPromiseAt c hc.le) := by
  let s := Classical.choice (exists_axesExponentSchedule hc h)
  let R := concreteIntegerTargetUniformReductionCertificate c hc.le s
  exact isNPHard_of_integerTargetUniformReductionCertificate c hc.le R
```

The rise of AI in Mathematics happily coincides with the rise of general machine-checkable math proofs and especially **Lean**: https://lean-lang.org/

Lean itself is not AI; it is a programming language with a carefully written logic kernel and libraries that formalize a lot of mathematics. Standard crypto things like Game-based proofs are still evolving (HOPSCOTCH [10] and VCVio [26].)

SECTION

002

HAWK: THE LEE SEDOL MOMENT

2016: ALPHAGO -VS- GO GRANDMASTER LEE SEDOL

HAWK – A PQC SIGNATURE SCHEME

HAWK [4, 9] was a NIST PQC Signature standardization (“On-Ramp”) candidate until July 29, 2026. Many saw HAWK as a successor algorithm to Falcon [11] (FIPS 206 / FN-DSA), although it is *based on different security foundations*.

Attractive: Fast signing & verification, 555-byte signatures, no floating point.

> QUOTE FROM NIST IR 8610, MAY 2026 [1]
>
> *NIST selected HAWK for the third round due to its strong performance and because its implementation only needs integer arithmetic operations. NIST encourages further analysis of HAWK’s security assumptions, particularly the smLIP problem within the specific structure of cyclotomic number fields.*

ANTHROPIC'S ATTACK ON HAWK

On **2026.07.28**: Z. Straznickas and S. A. Weis from Anthropic released: *"HAWK-n Key Recovery Reduces to SVP in Dimension n/2 + 1"* [25].

A polynomial reduction from HAWK-n to poly(n) exact Shortest Vector Problem (SVP) oracle calls in dimension $n/2 + 1$ (here n is the ring degree). With SVP, it is still exponential, but HAWK v1.1's BKZ analysis attacked a rank-$2n$ lattice.

> "I KNOW THIS!"
>
> The attack combines two well-known cyclotomic involutions to expose a previously hidden HAWK automorphism as a shortest vector in a public rank-n cocycle lattice, then applies Ducas reduction [8] and van Gent–Pulles descent [12]: *Most ingredients were well known to the HAWK Team*.

HAWK SECURITY ASSUMPTIONS

HAWK's security assumptions were *inspired* by the Lattice Isomorphism Problem (LIP). Over time, HAWK's proofs became more rigorous, but the authors openly set concrete parameters by cryptanalysis, not by the proofs:

Layer	2022 ASIACRYPT Hawk	2025 2nd Round Hawk v1.1
Key recovery	Power-of-two cyclotomic smLIP; concrete Gaussian keys outside the worst-to-average proof.	Same smLIP; binomial keys also outside that proof.
Unforgeability	omSVP reduction; concrete parameters more aggressive than the proof requires.	Game revised for a pseudo-automorphism; concrete omSVP remains solvable.
Parameters	BKZ estimates for unusual-SVP and approximate-CVP.	Expanded BKZ costing, including dimensions-for-free.

HAWK TIMELINE

- **2022.12.08**: Ducas, Postlethwaite, Pulles, van Woerden. *Hawk: Module LIP Makes Lattice Signatures Fast, Compact and Simple* [9]. ASIACRYPT '22.
- **2026.05.14**: Hawk selected to 3rd round of NIST PQC Signature On-Ramp.
- **2026.06.25**: Nelson, Limbrey, Ling, Mendelsohn. *Cryptanalysis of HAWK: a Guessing Game* [18]. Authors conjecture a classical polynomial-time attack.
- **2026.07.05**: Saarinen. *HAWK "Guessing Game" is not Polynomial-Time* [22]. AI-Assisted implementation attempt/analysis. ("It's okay!")
- **2026.07.28**: Straznickas, Weis. *HAWK-n Key Recovery Reduces to SVP in Dimension n/2 + 1* [25]. – Anthropic's AI cryptanalysis announcement.
- **2026.07.29**: HAWK Withdrawn from the NIST Competition by its authors.

30 JULY 2026 – “LEE SEDOL” (BUT CONCLUSIONS VARY)

REACTIONS: CONG LING ON NIST PQC FORUM

Dear Colleagues,

This is a historic moment in cryptology: AI has beaten humans at cryptanalysis. We would like to share a few thoughts with the PQC community based on our own experience.

There are other dimension-reduction attacks on HAWK in literature. At Eurocrypt 2026, we proposed a dimension-halving algorithm for quaternion Ideal-SVP, which potentially implies that HAWK-n key recovery can be reduced to (cyclotomic) Ideal-SVP in dimension n.

Later, we proposed another guessing attack based on advanced number theory, under a few heuristic assumptions. However, one of these heuristics was soon found to be invalid by colleagues with the assistance of AI.

Apparently, we were outpaced by AI in both cases. A couple of lessons we have learned are: (a) humans can make mistakes; and (b) humans are slower than AI. Beyond cryptanalysis itself, the use of AI to check mathematical proofs is becoming an increasingly serious issue.

One can even imagine a crisis in cryptography: in the worst-case scenario, all human-designed cryptosystems are broken by AI. Whether or not that happens, a new era – Post-AI Cryptography – has arrived.

TWO KINDS OF "HARD PROBLEMS"

1. **Computationally hard problems**, instances of which are hard for computers to solve. For example: SAT $\sim$ exponential $\sim$ SVP, SD, MQ, $\cdots$
2. **Hard theory problems**, instances of which humans have hoped to solve/settle by pen & paper, but failed: P $\neq$ NP, Riemann Hypothesis, $\cdots$

2026: Computers / AIs are getting better at solving problems traditionally in (2).

The hardness of (1) is studied in theoretical computer science and cryptography, but assessment of hardness of (2) has been much more subjective.

Many gaps remain in our "proofs of security/hardness" for (1). We can hope that AI-aided advances in (2) will help fill those gaps. **This is post-AI cryptography.**

SECTION

003

NGCC: CHINA’S PQC & HASH ALGORITHMS

NGCC: CHINA'S OPEN PQC & HASH COMPETITION

Submissions by Family & Category:

Family	Sign	KEM	KEX	Hash	Tot
Lattice	11	25	7	–	43
Code	5	15	1	–	21
Multivariate	6	–	–	–	6
Isogeny	3	1	1	–	5
Hash-based sig.	4	–	–	–	4
VOLE-in-the-head	4	–	–	–	4
Group action	1	–	–	–	1
Symmetric	–	–	–	35	35
Total	34	41	9	35	**119**

Run by ICCS (Institute of Commercial Cryptography Standards). Follows the NIST model: public calls, fixed C API, KATs, open mailing lists.

Announced Feb 2025; calls Oct 2025; submissions Jun 30, 2026.

119 candidates (NIST PQC: 69 in 2017, SHA-3: 51 in 2008).

NGCC: ONE CRYPTOGRAPHER, TOOLED-UP AGENTS

I have just a typical developer AI set up: OpenAI Codex (∼$200/month) and Claude Code (∼$100/month) on a 28-core cloud server, isolated Linux accounts, no sudo, no git write access. About 2.01 billion tokens in Sep 21–27.

AI uses tools: Python, SageMath, Rust, C, Lean, SAT solvers, lattice estimator..

Reference Literature: IACR ePrint, arXiv, proceedings (I have a large stash.)

Cross-vendor review ping-pong: Claude often reviews Codex and vice versa.

Source-only harness: 87,514 submitted files, including 471 prebuilt executables. Rule: *nothing shipped in a submission is ever executed*. Day one: 577 instances of 117 candidates built, 545 (94.5%) test their KATs byte for byte.

NGCC: GENERAL SECURITY AUDIT ITEMS

01 **Safety** Static-check submitted reference implementations (nothing very malicious found). Never execute shipped binaries, scripts or objects; build with our Makefiles, then isolate and time-limit tests.

02 **Evidence** Cite claim and source; reproduce a *witness* with a control. Tag research leads and proof gaps separately.

03 **Spec ↔ code** Compare the algorithm spec and its implementation: parameters, API headers, and Known-Answer Tests (KATs).

04 **Entropy Audit and Sensitivity Analysis** Trace DRBG use and randomness in keys, salts, nonces, and challenges, and internal variables.

PERHAPS THE BIGGEST “EXPERT” CONTRIBUTION

BRIEFS AND CHECKLISTS

KEY ENCAPSULATION

Mutate ciphertexts. Check full re-encryption, rejection masks, and fallback binding to (pk, ct); probe FO failure/plaintext oracles.

SIGNATURES

Mutate message, signature, and length. Check verifier bypass, non-canonical encodings, and nonce or challenge entropy.

HASHES

Probe padding, zero extension, cross-digest relations, long-message bounds, and structural limits.

TIMING SIDE-CHANNELS

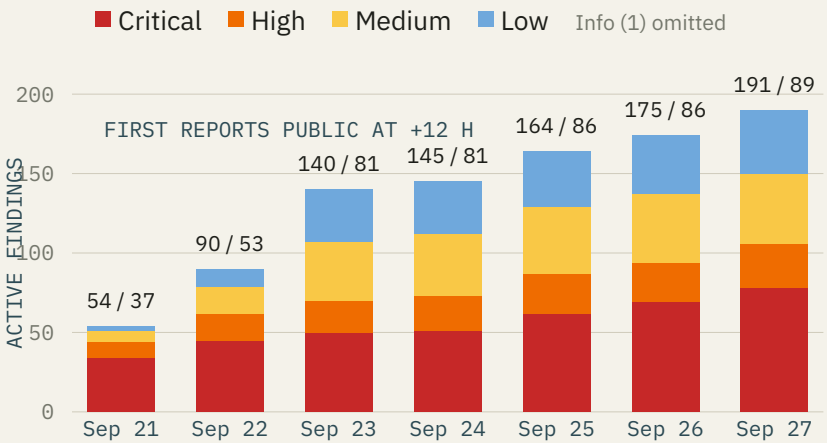
Sensitive (secret) variables are not used for branches, table lookups, non-constant time instructions.

NGCC: VULNERABILITY CLASSIFICATION

Critical	A violation of primary NGCC security requirements. *EUF-CMA (signatures), IND-CCA2 (KEMs), collisions, preimages,..*
High	A serious failure or well-supported attack path, short of a complete primary break.
Medium	A security-relevant defect or leak without demonstrated recovery or forgery.
Low	Narrow robustness, availability, or API impact; no cryptographic break shown.
Info	A tracking or out-of-model observation, without a candidate-specific break.

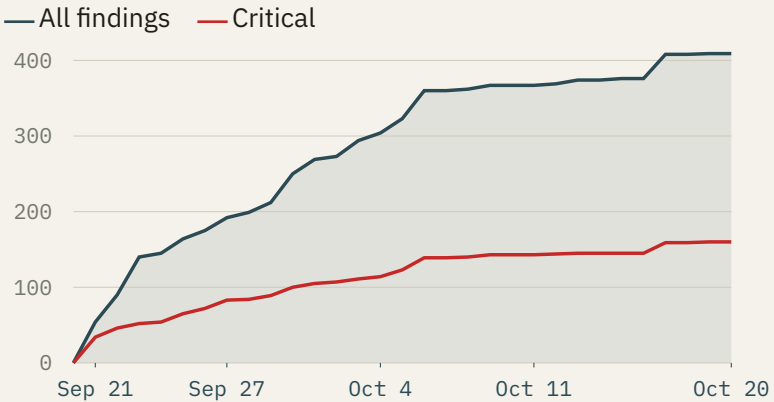
CUMULATIVE, LABELS: FINDINGS / REPORTS [21]

NGCC: 191 FINDINGS IN FIRST SEVEN DAYS



ACTIVE FINDINGS DATED SEP 21–OCT 20, 2026

NGCC: THE FIRST MONTH



OCT 20

409

active findings

160

Critical

110

candidates with findings

Current reports include one first-week finding added after the paper snapshot.

NGCC: FINDINGS OF THE FIRST WEEK (SEP 21–27)

	Scope			Category				
Severity	**Des.**	**Impl.**	**S.-c.**	**Sign**	**KEM**	**KEX**	**Hash**	**Total**
Critical	32	46	0	35	24	5	14	78
High	14	13	1	11	12	2	3	28
Medium	24	9	11	14	18	4	8	44
Low	6	17	17	9	13	2	16	40
Info	0	1	0	0	0	0	1	1
Total	76	86	29	69	67	13	42	**191**
My project + AI	24	59	27	30	45	12	23	**110**
Others + AI	10	9	1	15	5	0	0	20
Human / unknown	42	18	1	24	17	1	19	61

Status: 167 Confirmed, 10 Probable, 10 Leads, 4 Proof gaps.

NGCC: THE CRITICAL BUG ZOO ..

- **Verifier bypass** UVW returns success on failed signatures; any forgery passes.
- **Broken FO check** Amoeba compares only every fourth ciphertext byte; tampering leaks the secret key.
- **Randomness misuse** Galas and VDOO skip the RNG and repeat keys; 14 schemes reset the DRBG as a XOF or KDF.
- **Missing challenge binding** Chinith omits $\tilde{a}_0$ from the challenge; anyone can sign under any public key.
- **Index bugs** CEDRUS+C uses only tree 0 (four few-time keys); MORNING-ATLAS repeats masks, exposing the key after two signatures.
- **Bits vs. bytes** CreTAKE's 512-bit secret has only 64 bits of entropy.
- **Hash padding** Mixed bit ordering makes the empty string collide with seven zero bits.
- **Ciphertext aliases** OAEP-NTRU accepts alternate ciphertext bytes for the same key: a one-query IND-CCA break.

NGCC: DESIGN BREAKS FROM THE FIRST WEEK

Here the security issue is in the **specification / design**. It is often difficult to draw a line between a “design oversight”, and a “new cryptanalytic attack.”

Through day four, most **design attacks** came from human researchers (some AI-assisted): Tins witness by $\mathbb{F}_2$ elimination; Facto-DSA-128 forgery in 6.4 s; full-round MoFang hash collisions (Tsinghua).

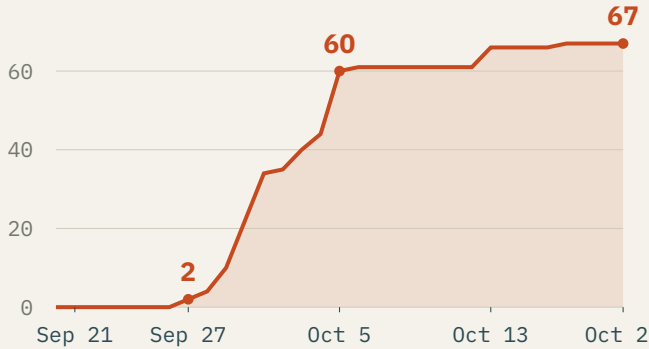
From day five, Codex-assisted signature breaks by Martin Feussner (Bergen): key recovery or forgery for Shuttle, Sigurd, ATLAS-128, Lynxer, SQIsignTriangle.

Of the 81 findings by others, 20 disclosed AI: OpenAI models (10), Chinese Doubao and Xuanzhi CryptoLLM (2), unnamed (8). Later a lot of GLM 5.3.

Still human: Scope, evidence, disclosure, and the decision to publish.

NGCC: THE CHINESE WAVE

Cumulative findings with named model assistance



OCT 13 BATCH

□□□□□□
□□□□□

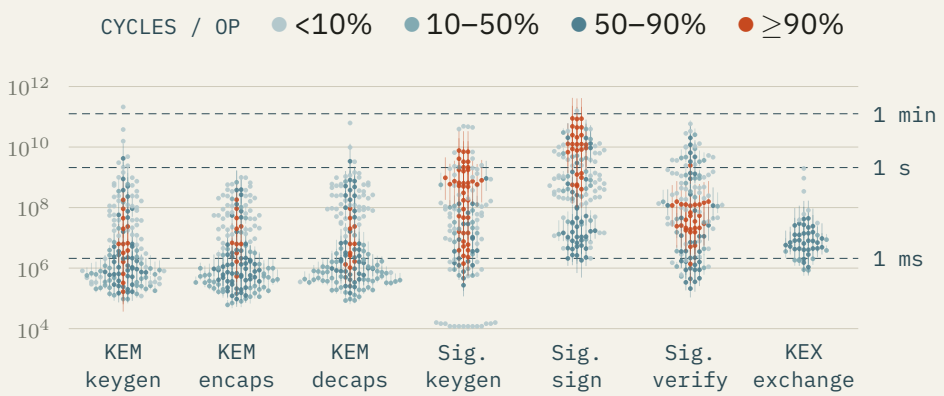
11 .md reports
162 observations

⇓

5 new IDs
3 findings + 2 Info

Credited to: Doubao, Xuanzhi, GLM 5.3, kimi-k3. Oct 13: other claims overlapped or did not meet criteria.

NGCC: PLACEHOLDER HASHES DOMINATE [21]



403 / 959 operations spend at least half their cycles in ICCS placeholder helpers.

SECTION

004

A BAD YEAR FOR MCELIECE

(ULTIMATELY THE WORST YEAR SINCE 1978)

2026: THE YEAR OF MCELIECE

Classic McEliece was standardized in ISO/IEC 18033-2:2006/Amd 2:2026 [17].

- **2026.02**: Pfaffian modeling + TII-210 break (Bardet-Lemoine-Tillich [2]).
- **2026.02**: Higher-Order Vanishing / distinguisher (Hemmert-Wiemers [16]).
- **2026.06**: Rank-2 Subexp. (Briaud-Lemoine-Randriambololona-Tillich [5]).
- **2026.07**: Higher-Order Vanishing + TII-248 break (Hemmert [15]).
- **2026.07**: HOVER + TII-252 break (Saarinen + AI [23]).
- **2026.08**: Holdout Quasipolynomial (Ghoshal-Ishai-Jain-Sun [13]).
- **2026.08**: Holdout costing + TII-253 break (Weis + Claude [27]).
- **2026.09**: Holdout costing + TII-254 break (Saarinen + AI [20, 24]).

FAVOURABLE REGIMES FOR STRUCTURAL ATTACKS

OBSERVATIONS

- These attacks are all conditional on parameter ranges (or even keys!)
- There is no single "input size" security parameter for asymptotics.
- A distinguisher does not directly imply decryption or OW-CPA break.

Attack	**Type**	**Indicative Conditions** *(very rough)*	**Asymptotics**
Pfaffian [2]	key recovery	Very high rate; square defect visible	poly in (m, r)
HOV [16]	distinguisher	Random HOV kernel dies; Goppa not	–
HOV [15]	key recovery	Clean Frobenius-split HOV kernel	–
Rank-2 [5]	key recovery	Rank-2 relation at low solving degree	subexp $2^{o(r)}$
Holdout [13]	key recovery	Kernel survives past Hermite threshold	qpoly $n^{O(\log n)}$

TII CHALLENGES

The Technology Innovation Institute (UAE) ran a McEliece challenge in 2023-24.

TII offered cash prizes (the reward for TII-70 and above was $ 10,000). Too bad it has ended – the new AI algorithms would have paid for a lot of AI tokens :-)

Note: The TII work factor label is for a "dumb" brute force attack on (n, m, t): Goppa-polynomial search $\times$ support-set search $\times$ $O(n^3)$ candidate verification.

$$\lambda_{\mathrm{TII}} = \log_2 \left(I(t, 2^m) \cdot \binom{2^m}{n} \cdot n^3 \right)$$

Varied (n, m, t) helped motivate research into different parameter regimes.
https://github.com/ElenaKirshanova/tii_decoding_challenge

HOVER: A HIDDEN FIELD INSIDE THE PUBLIC KEY

HOVER (Higher-Order Vanishing Endomorphism Recovery) is an extended variant of the HOV Key Recovery attack [15, 16], discovered via our AI experiments.

Idea: HOV yields an m-dimensional matrix space from a public key. Useful combinations lie along m secret directions, permuted cyclically by squaring.

Rather than search for those directions, ask which binary matrices act on the coefficients without carrying any relation outside the relation space:

$$\mathcal{E} = \{B \in \mathbb{F}_2^{m\times m} : B \text{ sends every relation to a relation}\} \;\cong\; \mathbb{F}_{2^m}.$$

Linear algebra finds $\mathcal{E}$. When derivative images split and their kernels are distinct, a field generator's eigenlines are exactly the hidden directions.

HOVER: NEW TII KEY RECOVERY RECORDS

HOVER turns out to be quite effective (against specific low-degree parameters.) Native multi-threaded implementation. Verified secret keys with official scripts.

Challenge	m	r	n	Wall Time	Memory	System
TII-252	10	11	1008	6 min 15 s	2.7 GiB	12-core laptop
TII-246	10	11	1009	3 min 11 s	2.7 GiB	12-core laptop
TII-240	10	11	1010	3 min 06 s	2.7 GiB	12-core laptop
TII-213	9	10	496	6 h 45 min	144 GiB	28-vCPU server
TII-129	9	9	509	4 h 30 min	80 GiB	28-vCPU server

Evidence (secret keys): https://github.com/mjosaarinen/tii-solved

Note: TII-83 was the “highest” challenge broken by the May 2024 deadline.

HOLDOUT / GIJS: THE MISSING COLUMN IS FORCED

Over $\mathbb{F}_{2^m}$, the public columns are scaled points $y_i = \lambda_i p(\alpha_i)$ on a hidden polynomial curve, with $\deg p_j \le D = n - 2t - 1$.

Hold out y_τ. Find a nonzero homogeneous polynomial h of degree d whose Hasse derivatives of order $< s$ vanish at every other column. Then $\phi(Z) = h(p(Z))$ has degree at most dD and at least $s(n-1)$ zeros, counting multiplicity:

$$s(n-1) > dD \quad \Longrightarrow \quad \phi \equiv 0 \quad \Longrightarrow \quad h(y_\tau) = 0.$$

There are still enough coefficients to find such an h; a random held-out column has no reason to satisfy the last equality. For $d = s + 1 = \Theta(\log n)$ this gives a quasipolynomial distinguisher [13].

HOLDOUT / GIJS: AI DISCLOSURE [13]

AI disclosure. Our main results emerged from the failed PIR attempt described above and were obtained through traditional human trial and error, without the use of AI. Still, AI was instrumental for many aspects of this work. OpenAI Codex (ChatGPT-5.6 Sol) was used to verify the results, analyze and improve concrete efficiency, draft some of the proofs, design and implement experiments, and help with literature review and other aspects of writing. Amazon Quick and Google AI harness were used for extra sanity checks.

AI played a more significant role in developing our key-recovery algorithm. The algorithm, inspired by the Sidelnikov–Shestakov algorithm, was proposed by the authors. ChatGPT-5.6 Sol then helped with the routine but tedious derivations of the structure of the interpolation coefficients. It also helped us quickly run the experiments measuring nullities, which were important for guiding the choice of parameters and gaining confidence in our conjectures.

Most importantly, an audit by ChatGPT identified a dimension gap that made our initial algorithm fail for binary Goppa codes and required changing the parameters. ChatGPT also helped run experiments to confirm the structure of the required interpolation coefficients for Goppa codes. We sent the resulting algorithm to Steve Weis from Anthropic, who was able (with the help of Claude) to successfully run our algorithm on small binary Goppa instances.

The authors verified the correctness and originality of all content.

HOLDOUT: COST AGAINST CLASSIC MCELIECE

Weis's key recovery: the interpolation relations also encode the hidden support. After removing local jet relations, their top-order blocks satisfy $V_{2,j} = \alpha_j^{s'-1} V_{1,j}$ (up to normalization and Frobenius conjugacy). Read off the α_j, recover the multipliers, and obtain an equivalent decoding key [27].

Parameter set	NIST Req [19]	ISD [3]	GIJS [13]	Weis [27]	Saarinen [20]
mceliece348864	143	151	114.2	94.3	126.8
mceliece460896	207	191	120.3	99.3	145.2
mceliece6688128	272	257	124.0	102.4	137.5
mceliece8192128	272	287	124.0	102.4	137.5

TII-253 ($m = 8, t = 9, n = 214$): two sparse kernels, ≈ 700 core-hours each [27].

SPECIAL HOLDOUT: TII-254 KEY RECOVERY

TII-254 $(m, t, n) = (8, 12, 223)$: hardest solved challenge so far [24].

I ran a human-directed, AI-operated algorithm improvement loop for 8 days.

TII-254-specific, 13-step process: Degree-6 holdout system; remove 4.8M provable "nuisance" directions (20.3M $\to$ 15.5M). Solve at two anchor columns: two 121-dim kernels. Their pair core splits as $80 = 64 + 16$; the common 64 is noise, and the $16 \cong \mathbb{F}_{2^8}^2$ has the support locators as its $\mathbb{F}_{2^8}$-lines.

Stage (per anchor)	Time	Hardware
Block Wiedemann sequence (width 512)	13.6 h	$1\times$ NVIDIA GH200
Polynomial-matrix basis	0.8–5.1 h	12–64 CPU threads
Kernel reconstruction	$\approx$2 h	$8\times$ GH200

Bundesamt
für Sicherheit in der
Informationstechnik

Deutschland
Digital•Sicher•BSI•

Hinweise zu aktuellen Entwicklungen um Classic McEliece

Stand: 29.09.2026

SEP 29, 2026: CURTAINS ON CLASSIC MCELIECE

Since 2020, Germany's cyber authority BSI has recommended Classic McEliece in its Technical Guideline TR-02102-1.

The January 2026 version still stated: *"The underlying mechanism, which is instantiated with binary Goppa codes, is about as old as the RSA mechanism and is therefore considered conservative and very thoroughly analysed."* [6]

On September 29, 2026 BSI revised their guidance:

"Classic McEliece should currently no longer be used in new developments or when planning new cryptographic applications." [7]

The BSI Classic McEliece security advisory [7] cites the same 2026 works as we did here [2, 5, 13, 15, 16, 27], including my works [23, 24].

SECTION

005

QUICK CONCLUSIONS

THE PROOF CHECKING BURDEN

Peter Scholze on the Liquid Tensor Experiment (June 2021):

> *Question: So, besides the authors of course, who understands the proof now?*
>
> *Answer: I guess the computer does, as does Johan Commelin.*

This was the situation with advanced mathematics already **before** advanced AI.

2026: Full versions of some conference papers are 80 pages with 50-page proof appendices that reviewers were not required to read. 30 pages are published.

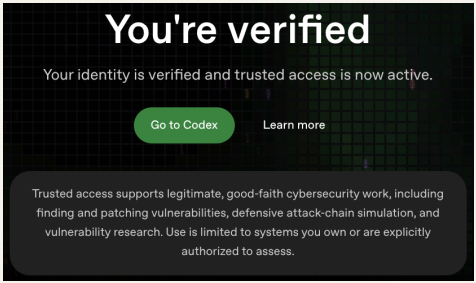
Near Future: AI allows more proof coverage – which should be a *good thing*. I'd predict conferences that already require implementation artifacts also to require machine-checkable proof artifacts (if the proof is a key contribution.)

REMINISCENT OF USG’S EFFORTS IN 1990S TO “CONTROL CRYPTOGRAPHY PROLIFERATION”

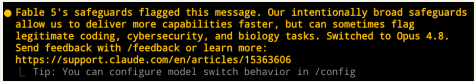
Claude

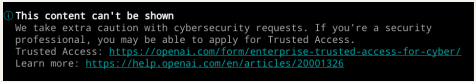
Hello,

Thank you for submitting your application for the Cyber Verification Program. Upon reviewing the details of your submission, we have adjusted the safeguards applied to your account to allow for the cyber use cases you described.



.. but very often, when working on cryptography research ..





MY OWN MODEST CONCLUSIONS

The way mathematics & engineering is done has changed radically in a year.
Learning and adaptation required for many. Curriculum rethink at Universities.

AI can't find bugs that aren't there – security improves.
Software patching is currently at an all-time high (cryptography not immune).

AI helps prove or disprove statements that humans have not been able to.
Some of the undiscovered theory and security proofs to defend against "Post-AI cryptanalysis" may turn out to be very elegant. But some of it may be so complex it is basically beyond human comprehension. ***Yet those proofs are necessary!***

Powerful, open mathematics and cryptography models are needed.
Open science needs to resist corporate and government control.

END OF TALK!

QUESTIONS

https://mjos.fi/doc/20261021-cwi-math-consequences.pdf

REFERENCES (1)

[1] Gorjan Alagic, Maxime Bros, Pierre Ciadoux, Quynh Dang, Thinh Dang, John Kelsey, Jacob Lichtinger, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Hamilton Silberg, Daniel Smith-Tone, and Noah Waller. “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, p. 38. DOI 10.6028/NIST.IR.8610.

[2] Magali Bardet, Axel Lemoine, and Jean-Pierre Tillich. “An Attack on the CFS Scheme and on TII McEliece Challenges”. In: *Advances in Cryptology - CRYPTO 2026 - 46th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-20, 2026, Proceedings, Part IV*. Ed. by Nadia Heninger and Mike Rosulek. Vol. 16803. Lecture Notes in Computer Science. Springer, 2026, pp. 380–412. DOI 10.1007/978-3-032-35398-6_13. URL: https://eprint.iacr.org/2026/430.

REFERENCES (2)

[3] Daniel J. Bernstein and Tung Chou. “CryptAttackTester: high-assurance attack analysis”. In: *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part VI*. Ed. by Leonid Reyzin and Douglas Stebila. Vol. 14925. Lecture Notes in Computer Science. Full version is available as IACR ePrint Report 2023/940. Springer, 2024, pp. 141–182. DOI 10.1007/978-3-031-68391-6_5. URL: https://eprint.iacr.org/2023/940.

[4] Joppe W. Bos, Olivier Bronchain, Léo Ducas, Serge Fehr, Yu-Hsuan Huang, Thomas Pornin, Eamonn W. Postlethwaite, Thomas Prest, Ludo N. Pulles, and Wessel van Woerden. “HAWK version 1.1 (February 5, 2025)”. Feb. 2025. URL: https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-2/spec-files/hawk-spec-round2-web.pdf.

[5] Pierre Briaud, Axel Lemoine, Hugues Randriambololona, and Jean-Pierre Tillich. “A Heuristic Subexponential Attack on the McEliece Cryptosystem”. Cryptology ePrint Archive, Paper 2026/1232. 2026. URL: https://eprint.iacr.org/2026/1232.

REFERENCES (3)

[6] BSI. “Cryptographic Mechanisms: Recommendations and Key Lengths”. Technical Guideline TR-02102-1. Bundesamt für Sicherheit in der Informationstechnik, Jan. 2026, p. 92. URL: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.pdf.

[7] BSI. “Notes on recent developments concerning Classic McEliece”. Security Advisory. Sept. 2026. URL: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Crypto/Notes_Classic_McEliece.pdf.

[8] Léo Ducas. “Provable lattice reduction of $\mathbb{Z}^n$ with blocksize $n/2$”. In: *Des. Codes Cryptogr.* 92.4 (2024), pp. 909–916. DOI 10.1007/S10623-023-01320-7.

[9] Léo Ducas, Eamonn W. Postlethwaite, Ludo N. Pulles, and Wessel P. J. van Woerden. “Hawk: Module LIP Makes Lattice Signatures Fast, Compact and Simple”. In: *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*. Ed. by Shweta Agrawal and Dongdai Lin. Vol. 13794. Lecture Notes in Computer Science. Springer, 2022, pp. 65–94. DOI 10.1007/978-3-031-22972-5_3. URL: https://eprint.iacr.org/2022/1155.

REFERENCES (4)

[10] Stefan Dziembowski, Grzegorz Fabiański, Daniele Micciancio, and Rafał Stefański. “Game Hopping in Lean”. 2026. arXiv: 2608.06261 [cs.CR]. URL: https://arxiv.org/abs/2608.06261.

[11] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. “Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU (Specification v1.2 — 01/10/2020)”. Oct. 2020. URL: https://falcon-sign.info/falcon.pdf.

[12] Daniël M. H. van Gent and Ludo N. Pulles. “HAWK: Having Automorphisms Weakens Key”. In: *IACR Commun. Cryptol.* 2.2 (2025), p. 20. DOI 10.62056/A3QJP2W9P.

[13] Ashrujit Ghoshal, Yuval Ishai, Aayush Jain, and Nuozhou Sun. “Quasipolynomial Cryptanalysis of the McEliece Cryptosystem (or: PIR Meets McEliece)”. Cryptology ePrint Archive, Paper 2026/1630. Aug. 2026. URL: https://eprint.iacr.org/2026/1630.

[14] Aron Gohr. “Improving Attacks on Round-Reduced Speck32/64 Using Deep Learning”. In: *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Vol. 11693. Lecture Notes in Computer Science. Springer, 2019, pp. 150–179. DOI 10.1007/978-3-030-26951-7_6. URL: https://eprint.iacr.org/2019/037.

REFERENCES (5)

[15] Tobias Hemmert. “Key Recovery for the McEliece Cryptosystem Using Higher-Order Vanishing”. Cryptology ePrint Archive, Paper 2026/1339. 2026. URL: https://eprint.iacr.org/2026/1339.

[16] Tobias Hemmert and Andreas Wiemers. “Distinguishing Goppa Codes Using Higher-Order Vanishing”. In: *Advances in Cryptology – CRYPTO 2026. 46th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17–20, 2026, Proceedings, Part IV*. Ed. by Nadia Heninger and Mike Rosulek. Vol. 16803. Lecture Notes in Computer Science. Springer, 2026, pp. 191–223. DOI 10.1007/978-3-032-35398-6_7. URL: https://eprint.iacr.org/2025/1661.

[17] ISO. “Information technology – Security techniques – Encryption algorithms – Part 2: Asymmetric ciphers – Amendment 2”. en. Standard ISO/IEC 18033-2:2006/Amd 2:2026. International Organization for Standardization, 2026. URL: https://www.iso.org/standard/86890.html.

[18] Ben Nelson, Joshua Limbrey, Cong Ling, and Andrew Mendelsohn. “Cryptanalysis of HAWK: a Guessing Game”. Cryptology ePrint Archive, Paper 2026/1318. June 2026. URL: https://eprint.iacr.org/2026/1318.

REFERENCES (6)

[19] NIST. “Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process”. NIST Call for PQC Standardization Candidates. Dec. 2016. URL: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf.

[20] Markku-Juhani O. Saarinen. “Bit Operation Cost of “Holdout” Key-Recovery Attacks Against Classic McEliece”. Cryptology ePrint Archive, Paper 2026/1786. Sept. 2026. URL: https://eprint.iacr.org/2026/1786.

[21] Markku-Juhani O. Saarinen. “Chinese NGCC Algorithms: The First Week of AI Cryptanalysis”. Cryptology ePrint Archive, Paper 2026/2266. Oct. 2026. URL: https://eprint.iacr.org/2026/2266.

[22] Markku-Juhani O. Saarinen. “HAWK “Guessing Game” is not Polynomial-Time”. Cryptology ePrint Archive, Paper 2026/1377. July 2026. URL: https://eprint.iacr.org/2026/1377.

[23] Markku-Juhani O. Saarinen. “HOVER: Higher-Order Vanishing Endomorphism Recovery”. Cryptology ePrint Archive, Paper 2026/1778. Aug. 2026. URL: https://eprint.iacr.org/2026/1778.

REFERENCES (7)

[24] Markku-Juhani O. Saarinen. “Two-Anchor Holdout/Hermite: Solving the TII-254 McEliece Key Recovery Challenge”. Cryptology ePrint Archive, Paper 2026/1986. Sept. 2026. URL: https://eprint.iacr.org/2026/1986.

[25] Zygimantas Straznickas and Stephen A. Weis. “HAWK-n Key Recovery Reduces to SVP in Dimension $n/2 + 1$”. Cryptology ePrint Archive, Paper 2026/1593. 2026. URL: https://eprint.iacr.org/2026/1593.

[26] Devon Tuma, Quang Dao, James Waters, Alexander Hicks, and Nicholas Hopper. “VCVio: Verified Cryptography in Lean via Oracle Effects and Handlers”. Cryptology ePrint Archive, Paper 2026/899. 2026. URL: https://eprint.iacr.org/2026/899.

[27] Stephen A. Weis. “Improving GIJS Key Recovery for Classic McEliece”. Cryptology ePrint Archive, Paper 2026/1984. Sept. 2026. URL: https://eprint.iacr.org/2026/1984.

[28] Emily Wenger, Mingjie Chen, Francois Charton, and Kristin Lauter. “SALSA: attacking lattice cryptography with transformers”. In: *Proceedings of the 36th International Conference on Neural Information Processing Systems*. NeurIPS ’22. New Orleans, LA, USA: Curran Associates Inc., 2022. DOI 9781713871088. URL: https://eprint.iacr.org/2022/935.