6TH XJTLU DING LAB POST-QUANTUM CRYPTOGRAPHIC MIGRATION WORKSHOP

ngcc.dev

Agentic Flows for Cryptanalysis and Implementation Security Assessment

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

20261129-ngcc.pdf

https://mjos.fi/doc/20261129-ngcc.pdf

TLP:CLEAR

CONTENTS

001 Intro: Post-AI Cryptography Era

002 China’s NGCC Competition

003 Interim Security Evaluation Results

004 On Agentic Flows used for ngcc.dev

SECTION

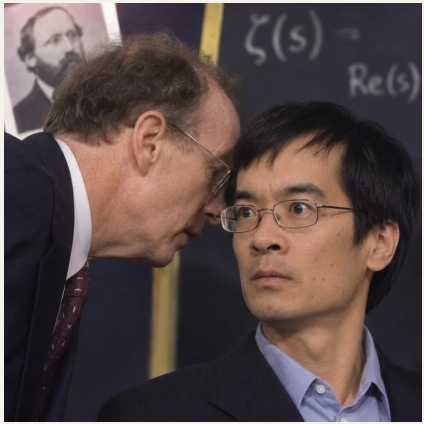
001

INTRO: POST-AI CRYPTOGRAPHY ERA

MARCH 2016: ALPHAGO VS. LEE SEDOL (9-DAN)



2026: RIEMANN HYPOTHESIS, ETC.



(Note: Not a real image.)

- Exponential Time Hypothesis? Gone. ✅
- Riemann Hypothesis? Probably done. ✅
- Classic McEliece. Gone. ✅
- Quasipolynomial Factoring / Discrete Log?
- Polynomial Quantum LWE?
- $P = NP$? Will it matter?

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 [4], SALSA LWE attacks from 2022 [8].

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 [6], 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.

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.
- The sandbox gives the models full SageMath, the Rust toolchain, Lean, and 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 [2] and VCVio [7]).

TWO KINDS OF "HARD PROBLEMS"

1. **Hard computational problems**: Instances of SAT, SVP, SD, MQ, ···
   (Seem exponentially hard even for quantum computers.)
2. **Hard "human" problems**: P = NP, Riemann Hypothesis, ···
   ("Lots of people have tried" – for decades, centuries.)

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

Generally, we want to prove (2) that a crypto algorithm is a hard instance (1). The hardness of (1) is studied in theoretical computer science, but the assessment of hardness of (2) – and some cryptography – has been more subjective.

Post-AI, we can have cryptography with better security evidence and proofs. Some machine-checked proofs may have to be beyond human comprehension.

SECTION

002

CHINA’S NGCC COMPETITION

NGCC: CHINA'S OPEN PQC & HASH COMPETITION

Run by **ICCS** (Institute of Commercial Cryptography Standards). Follows the U.S. 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).

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**

WHAT COUNTS AS BREAKING A SECURITY CLAIM?

ICCS asked submitters to propose 128-, 256-, and 512-bit security parameter sets. By convention, an attack that requires less effort is a “break”. Examples:

KEM IND-CCA2	Tell the encapsulated key from a random key. You may ask for decapsulation of other ciphertexts.
Signature EUF-CMA	After obtaining signatures on messages of your choice, produce a valid signature on a new message.
Hash Collision resistance	Find two distinct messages with the same digest. A 512-bit digest has a 256-bit classical collision target.

Example: A collision attack with 2^{200} hash evaluations would break the 256-bit claim. Such a computation is of course far beyond what anyone can do in 2026.

ROUND 1 CODE AND PLACEHOLDER HASHES

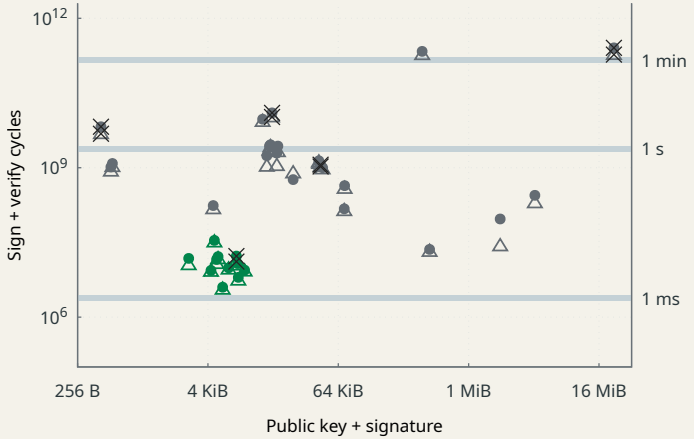
Every submission had to include a **portable ISO C reference implementation** and an optimized implementation for 64-bit PCs. Both had to use the ICCS API.

Public-key submissions had to use ICCS functions for hashing and extendable output (XOF). Both placeholders used SM3.

`pseudohash`	Produces a 512-, 768-, or 1024-bit digest.
`pseudoX0F`	Expands an input to the requested output length.

ICCS supplied these functions for correctness tests and preliminary timings. Later rounds would replace them with new hashes and XOFs.

DIGITAL SIGNATURES



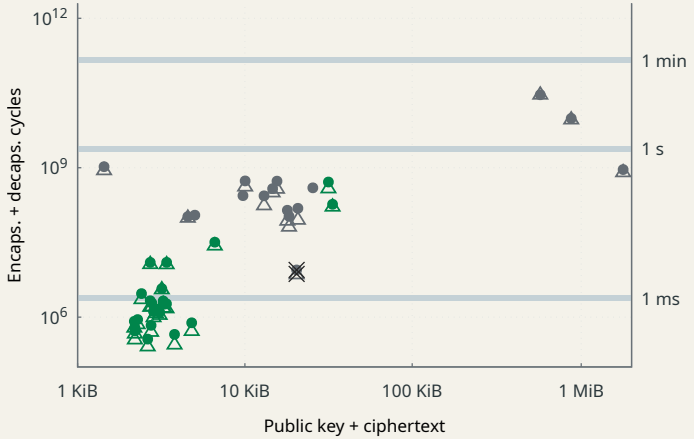
Lattice-based
Other families

● Intel 2.10 GHz
△ Arm 2.71 GHz

Cycles to sign and verify one message. The transfer includes the public key once.

× marks a failed known-answer test.

KEY ENCAPSULATION



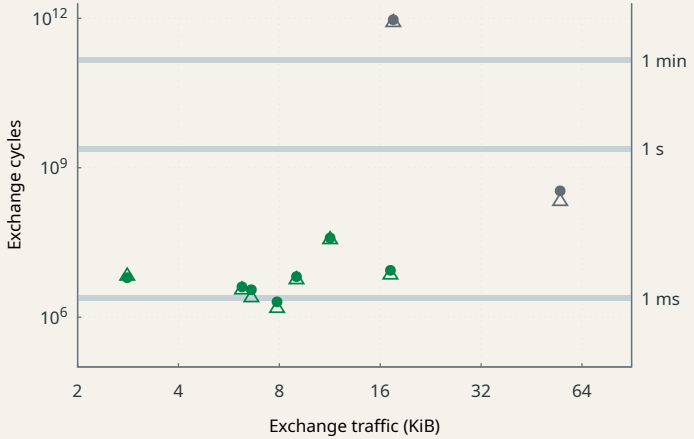
Lattice-based
Other families

• Intel 2.10 GHz
△ Arm 2.71 GHz

Cycles to encapsulate and decapsulate one key. The transfer includes the public key once.

× marks a failed known-answer test.

KEY EXCHANGE



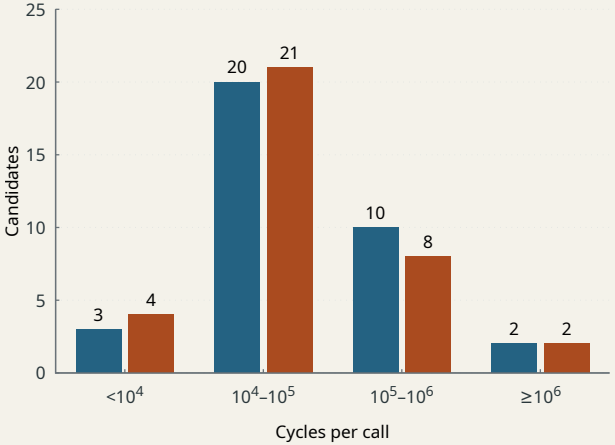
■ Lattice-based
■ Other families

• Intel 2.10 GHz
△ Arm 2.71 GHz

Total cycles and traffic for one complete exchange.

The shaded bands show equivalent times at the two CPU clocks.

HASHING A 1 KIB MESSAGE

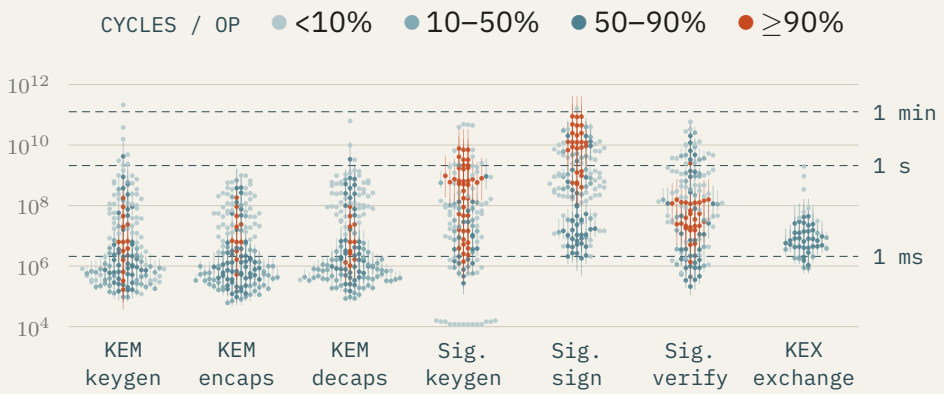


Intel 2.10 GHz
Arm 2.71 GHz

Each bar counts candidates by the cycles needed for one complete hash call.

Of the 35 candidates, 20 on Intel and 21 on Arm took 10^4 to under 10^5 cycles.

NGCC.DEV: PERFORMANCE TRIALS [5]



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

SECTION

003

INTERIM SECURITY EVALUATION RESULTS

NGCC.DEV: 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.

NGCC FINDINGS IN THE FIRST TWO MONTHS

86 of 119 candidates had at least one Critical finding.

	Scope			Category				
Severity	Design	Impl.	Side-ch.	Sign	KEM	KEX	Hash	Total
Critical	99	83	2	82	57	13	32	184
High	25	45	5	36	30	3	6	75
Medium	58	35	20	37	40	17	19	113
Low	21	133	30	61	65	15	43	184
Total	**203**	**296**	**57**	**216**	**192**	**48**	**100**	**556**
Authors + AI	56	144	44	85	86	24	49	244
Others + AI	73	53	8	64	50	16	4	134
AI undisclosed	74	99	5	67	56	8	47	178

488 Confirmed, 17 Probable, 24 Leads, 27 Proof gaps.

NGCC.DEV: 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.

LEAKY FIAT–SHAMIR CONSTRUCTIONS

In Fiat–Shamir signature constructions, the signer publishes a response that depends on the secret key. The response should hide information about the key. In schemes with aborts, the signer can discard a response and retry.

In **BiT** `sign-02-2`, **Shuttle** `sign-23-1`, **ATLAS** `sign-15-7` and **DARTS** `sign-08-2`, researchers found key-dependent patterns across valid signatures. In each case, they recovered an equivalent signing key: enough to sign a fresh message. The first three discoveries were explicitly reported as AI-assisted.

All of these have full-size forgery demonstrations. The consequence carried into key exchange: Loom `kex-05-3` used Shuttle and CreTAKE `kex-03-9` used BiT. Researchers demonstrated responder impersonation in both protocols.

ENUMERATION ATTACKS (SEARCH SHORTCUTS)

Insufficient entropy. At least 16 Confirmed Critical design findings identified a secret variable “too easy to guess”, invalidating their security claims.

Secret keys were sampled with too little entropy in 13 schemes.

In **Origami** `sign-18-1`, the hash length limits forgery resistance. In **MAMBA-Viper** `kem-21-2`, the seed limits session-key secrecy. s As a result, their 384- and 512-bit parameter sets have 2^{256} attacks.

Multi-target seed search shortcuts. Lynxer `sign-14-2` and ReSolveD-α `sign-21-1` (VOLE-in-the-head schemes) each offer hundreds of tweak search targets within a single signature. At the 256-bit level, Xiong and Wang estimated about 2^{248} evaluations for recovery in the affected sets.

PUBLIC KEY EXPOSES AN ALGEBRAIC STRUCTURE

In NSS-HQC `kem-26-1` and HEP-QC `kem-17-7`, public values have recognizable algebraic structure. The security proofs assume those values look random. The findings identify a gap in that argument.

Verification, decapsulation and decryption failures. Two designs omitted a source of noise in their decryption error estimates. Including it raised the estimated probability that a valid ciphertext fails to decrypt.

	Original estimate	Revised estimate
Amoeba-576 `kem-02-3`	2^{-130}	2^{-86}
Cheetah256 `kem-09-3`	2^{-176}	2^{-52}

STRUCTURAL AND LONG-MESSAGE ATTACKS ON HASH FUNCTIONS

Collisions. Structural findings lowered collision bounds for CHIME `hash-26-1` and CHAMP `hash-04-7`. On Oct 28, Sunan Wang published two messages with the same CHAMP-512 digest (`hash-04-6`).

Second preimages. For a fixed, very long message, three designs fell short of their 2^{1024} target. The estimated costs remain infeasible.

	uHash	JuziHash	QuantaSylva
Estimated work	2^{968}	2^{970}	2^{975}
Finding	`hash-05-2`	`hash-13-1`	`hash-24-2`

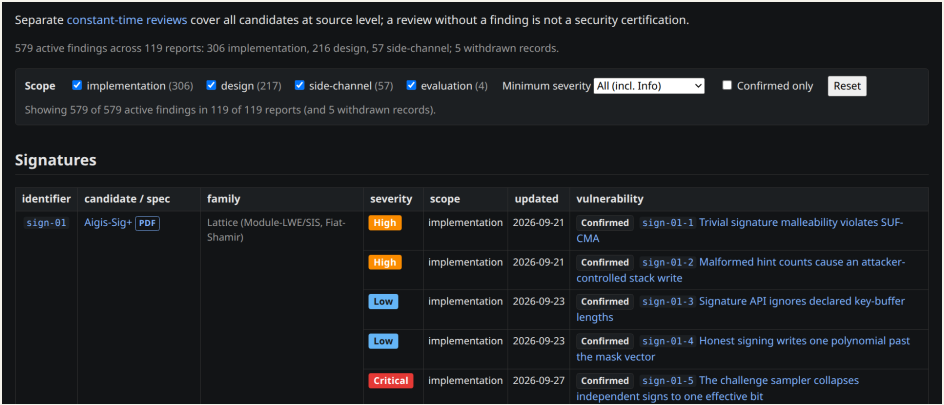
SECTION

004

ON AGENTIC FLOWS USED FOR NGCC.DEV

NGCC.DEV: TRACKING WEBSITE

We set up the website https://ngcc.dev to track vulnerabilities in NGCC candidates on September 21, the day of public availability of the candidates.



NGCC.DEV: DEVELOPMENT SETUP

Just a typical developer AI setup: OpenAI Codex, Claude Code subscriptions (total ∼$400 / month) on a 28-core cloud server with isolated Linux accounts.

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*. On day one, we built 577 instances of 117 candidates. Of those, 545 (94.5%) matched their KAT outputs byte for byte.

NGCC.DEV: 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, challenges, and internal variables.

PERHAPS THE BIGGEST “EXPERT” CONTRIBUTION

NGCC.DEV: 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.

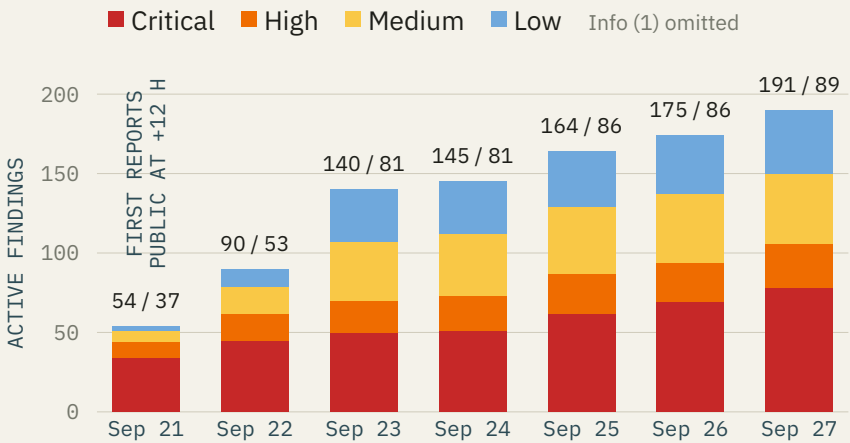
HASHES

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

TIMING SIDE CHANNELS

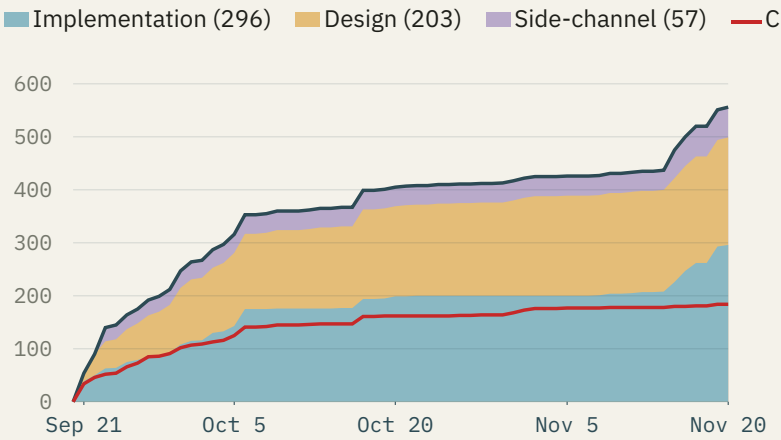
Sensitive (secret) variables are not used for branches, table lookups, or variable-time instructions.

NGCC.DEV: 191 FINDINGS IN FIRST SEVEN DAYS



BY FILING DATE, SEP 21–NOV 20, 2026

NGCC.DEV: THE FIRST TWO MONTHS



NOV 20

556

rated findings

184

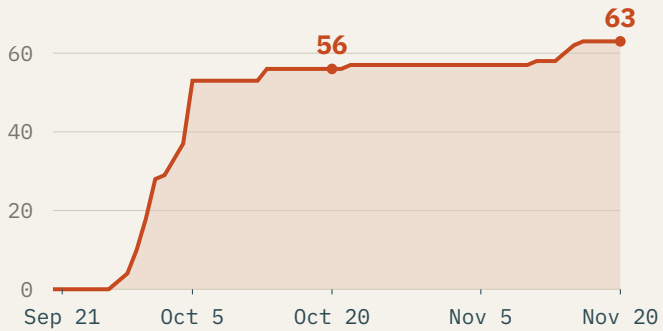
Critical

119

of 119 candidates with findings

NGCC.DEV: THE CHINESE OPEN-WEIGHT WAVE

Initial discoveries crediting a named Chinese model



OCT 13 BATCH

11 .md reports
162 observations

⇓

5 new IDs
3 findings + 2 Info

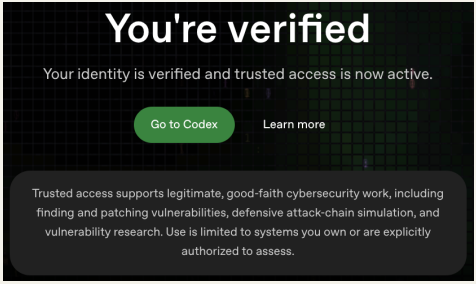
Of 63 Chinese-lab AI vulnerability findings, 56 named GLM-5.3 [3]. There were only seven new filings in month two. Five credited Shanghai AI Lab's group and two named a DeepSeek model.

NOTE ON REFUSALS BY CLOSED MODELS

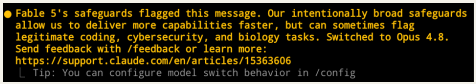
Claude

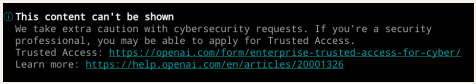
Hello,

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…





SAFEGUARDS CAN ALSO LIMIT EFFECTIVENESS

Safeguards can also be built to not refuse, but silently sabotage the results.

Unlike our interventions for cybersecurity, biology and chemistry, and distillation attempts, these safeguards will *not* be visible to the user. Fable 5 will not fall back to a different model. Instead, the safeguards will limit effectiveness through methods such as prompt modi□cation, steering vectors, or parameter-ef□cient □ne-tuning (PEFT). These interventions will not affect the vast majority of coding work. We estimate they will impact ~0.03% of traffic, concentrated in fewer than 0.1% of organizations. When these interventions are active, we expect them to have minimal behavioral impact on the model except to limit its effectiveness in developing frontier LLMs. Claude will still respond helpfully to user requests. We'll continue to improve the precision of our detection methods following the launch of this model.

From Section 1.5 of the initial Fable 5 system card, June 9, 2026 [1].

END OF TALK!

QUESTIONS



https://mjos.fi/doc/20261129-ngcc.pdf

REFERENCES (1)

[1] Anthropic. “System Card: Claude Fable 5 & Claude Mythos 5”. Initial June 9, 2026 revision. June 2026. URL: https://assets.claude.com/d00db56fa754a1b115b6dd7cb2e3c342ee809620.pdf.

[2] 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.

[3] GLM-5-Team. “GLM-5: from Vibe Coding to Agentic Engineering”. 2026. arXiv: 2602.15763 [cs.LG]. URL: https://arxiv.org/abs/2602.15763.

[4] 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.

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

REFERENCES (2)

[6] 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.

[7] 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.

[8] 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.