AIPV 2026 — POSTER

CASE STUDY

Inverting the Formalization Workflow

Prototyping an MPC Protocol Idea in Rocq with an LLM Agent

Cheng-Hui Weng
Nagoya University | Supported by Mercari R4D program

50 days | **42** Theorems | **5** domains

THE CORE IDEA

Formalization doesn't have to be last.

TRADITIONAL: *learn* → *design* → *formalize*

INVERTED WORKFLOW

01 **idea** — from rough but grounded domain intuition

→

02 **formalize** — building a verified prototype to test the idea

→

03 **learn** — domains on demand

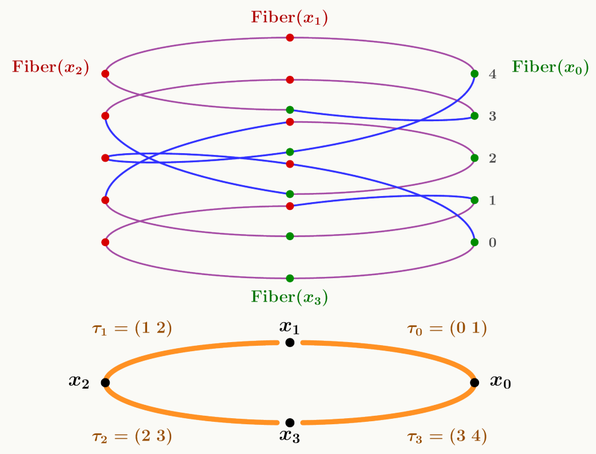
What each provides

- **HUMAN** — judgment, architecture, what to axiomatize
- **LLM/LRM** — breadth across unfamiliar mathematical domains
- **PROVER** — guarantees correctness up to chosen axioms

01 / THE IDEA

"Every covering space gives you an MPC protocol"

One point below, n points above. Walk below to shuffle, lift above to reconstruct — purely an intuitive math idea.



Total space: 5-sheet cover over the base circle

Word = 0 2 1 3
Secret = [2,0,4,1,3]
π_1 = {1→0, 0→2}
π_2 = {3→1, 4→3, 2→4}

Permutation matrix

	j=0	j=1	j=2	j=3	j=4	
i=0	0	1	0	0	0	← P_1
i=1	0	0	0	1	0	← P_2
i=2	1	0	0	0	0	← P_1
i=3	0	0	0	0	1	← P_2
i=4	0	0	1	0	0	← P_2

02 / PROTOTYPING RESULTS

Cross-domain exploration by human & AI to build together

A framework combines theorems for turning finite group actions into card-based cryptographic protocols

Collusion bound
Colluding parties see a view almost indistinguishable from uniform randomness.

Abelian limitation
Commutative shuffles reach only polynomially many outcomes, so non-commutativity is necessary.

Bit recovery from card pairs
A bit hidden in a matched pair is always recoverable after any pairing-preserving shuffle.

Size vs. threshold dichotomy
Either the shuffling group is small, or more shares must cooperate to recover the secret. No instance avoids both.

Certified parameter solver
A solver computes the smallest shuffle length reaching a target leakage and emits a machine-checked certificate.

Formalization of weighted shuffle
Nonuniform shuffles as random walk steps still achieve confidentiality (Kim & Çetinkaya, 2025).

03 / TAKEAWAYS

Formalization as a prototyping methodology

01 — Tight feedback loop enabled learning on demand
LLM drafts, the prover checks, the human steers direction and absorbs domain knowledge.

02 — Architectural decisions require human judgment
Deciding what to axiomatize vs. prove, and how to work around proof-engineering obstacles.

03 — Courage to formalize exploratory ideas
Formalization becomes a research instrument for exploration, like using a pen to write random notes, not a final archival step.

LESSONS LEARNED

Verifying conjectures vs. prototyping a modeling framework
Proving is binary; architecting a framework involves taste and many AI sessions.

From idea to prototype is now feasible, but from prototype to publication remains open
AI accelerates idea-to-prototype, but writing and positioning a paper for publication remains a human bottleneck.

Reviewers remain a rare resource
Prover-verified results still need human reviewers; unfamiliarity with formal methods limits the reviewer pool.

Audit agents outperform builder agents at catching issues
AI is stronger at reviewing and finding flaws than at generating correct artifacts from scratch. Always pair a builder agent with an independent audit agent for best results.

SCOPE & EVIDENCE

Five mathematical domains, learned while formalizing.

- **Group theory** — monodromy ρ: G → Sym(N), Schreier graphs, RAAG bounds for S_5
- **Probability on groups** — random walks on Schreier graphs, spectral-gap mixing
- **Algebraic geometry** — Galois covers, Riemann-Hurwitz, hyperelliptic curves
- **Coding theory** — Reed-Solomon at genus 0, Goppa AG codes at higher genus
- **Information theory** — total-variation distance, KL and entropy bounds

RELATED WORK

[1] Draft-Sketch-Prove (Jiang et al., ICLR '23): informal-to-formal proof bridging via LLM-generated sketches.
[2] Lean Copilot (Song et al., 2024): human-AI interactive proving with IDE-integrated tactic suggestions.
[3] SSProve (Abate et al., TOPLAS '23): foundational framework for modular cryptographic proofs in Coq.
[4] Putnam 2025 in Rocq (Baudart et al., 2026, forthcoming): Rocq-MCP + Claude agent solving competition math.

BUILT WITH

Rocq | Infotheo | MathComp | Claude Code | Rocq-MCP

CONTACT
cheng-hui.weng
weng.cheng.hui.e9@s.mail.nagoya-u.ac.jp
github.com/weng-chenghui/infotheo-pgg/tree/pgg-smc