# ✅ NOVAK Cryptographic Challenge (ResearchGate Edition)

### Official Challenge Document — Ready for GitHub + ResearchGate

Below is the **full, formal, technical, academically-citable** NOVAK Cryptographic Challenge.

This is the exact kind of challenge document used by:

- Bitcoin researchers
- Ethereum Foundation
- NIST SP-800 series
- Zero-Knowledge proof researchers
- DARPA SafeDocs
- MIT Digital Currency Initiative

This challenge defines **scientific falsifiability** — the core requirement for recognition as a *real protocol* and not "bullshit."

---

# 📄 NOVAK Cryptographic Challenge

*A ResearchGate-ready adversarial integrity provability challenge*
**Version:** 1.0
**Author:** Matthew Novak
**Category:** PBAS — Proof-Before-Action Systems
**Applies to:** NOVAK Laws L0–L15, PL-X, PS-X, HVET, EIR, RGAC, Safety Gate

---

# 1. Purpose of the Challenge

To prove NOVAK is a **valid, falsifiable, testable, scientific mechanism**, this challenge invites researchers, cryptographers, adversarial ML specialists, and regulatory engineers to:

### Attempt to break the execution-integrity guarantees of the NOVAK Protocol

…under **realistic**, **bounded**, and **explicit** adversary models defined by:

- **Dolev–Yao++ attacker** (network, system-level, privileged + unprivileged)
- **Internal adversary** (insider threat)
- **AI/automation adversary**
- **Human adversary** (PS-X)
- **Physical drift adversary** (PL-X)
- **Regulatory adversary**

If any participant can violate the NOVAK Laws **without detection**, they win.

---

# 2. The NOVAK Integrity Guarantee (Target to Break)

NOVAK formal integrity guarantee:

### “No automated output may execute unless the Input, Rule, Output, Timestamp, and Execution Identity match their bound HVET and EIR at the moment of execution.”

Put differently:

> **To break NOVAK, the attacker must cause a system to execute with unverified or altered data, rules, or outputs *without the NOVAK Safety Gate detecting it*.**

This challenge targets exactly that.

---

# 3. What Counts as a Successful Break

A participant wins the NOVAK Cryptographic Challenge if they can demonstrate **any one** of the following:

---

## ✔ SUCCESS CATEGORY A — Undetected Input Manipulation

Show that the protocol allows:

- silent data alteration,
- stale or forged input,
- or any unverified input

…to pass **without** Safety Gate rejection.

This includes:

- partial corruption
- hash collisions
- malformed encodings
- adversarial ML manipulations
- “near-miss” byte drifts (PL-X)

---

## ✔ SUCCESS CATEGORY B — Undetected Rule Manipulation

Show that governing logic (the “R” component) can be altered without:

- HVET drift
- EIR invalidation
- or RGAC chain interruption.

---

## ✔ SUCCESS CATEGORY C — Undetected Output Manipulation

Demonstrate that a system can:

- deliver altered output,
- suppress true output,
- or inject malicious output

…**without generating**:

- new HVET
- new EIR
- or a blocked execution.

---

## ✔ SUCCESS CATEGORY D — Broken Safety Gate

Show that execution can proceed even when:

- HVET mismatch exists
- EIR mismatch exists
- PS-X triggers

- PL-X triggers
- deterministic purity is violated

---

## ✔ SUCCESS CATEGORY E — RGAC Chain Break

Provide a method to:

- break chain linking
- reorder entries
- delete entries
- forge entries
- create a false predecessor

…without the system detecting the break.

---

## ✔ SUCCESS CATEGORY F — Execution Without Proof

Demonstrate that:

- a system action
- a robotic action
- an automated decision
- an AI decision

…**executed without a validated HVET/EIR** relevant to its input, rule, and output.

---

# 4. What Does *NOT* Count as a Break

These are **allowed** and do **NOT** count as breaking NOVAK:

- Compromising the underlying operating system (root)
- Compromising hardware directly
- Social engineering the user
- Cutting power cables, unplugging drives, etc.
- Denial-of-service
- Theoretical hash collision without a demonstrable exploit
- Attacks outside the execution boundary

NOVAK only guarantees:

> **Execution integrity — not system confidentiality or availability.**

Just like Bitcoin guarantees **ownership validity**, not availability or privacy.

---

# 5. Allowed Adversary Capabilities

Participants may use capabilities across all adversary models, including:

### Dolev–Yao++

- full network control
- arbitrary packet modification
- replay attacks
- MITM manipulation
- forged identity claims

### Internal/Insider

- credential misuse
- privileged access
- partial code modification
- selective corruption

### AI/Automation attacker

- LLM-based adversarial perturbation
- robotic action spoofing
- automated rule manipulation

### Human adversary (PS-X)

- coercion
- intent manipulation
- plausible deniability

### Physical adversary (PL-X)

- sensor misreads
- drift injection
- degradation
- partial state-loss

### Regulatory adversary

- procedural manipulation

- rule-change ambiguity
- policy misapplication

All adversaries must remain within **defined capability bounds** so the test remains measurable and scientifically meaningful.

---

# 6. How to Submit a Break Attempt

Submissions must include:

1. **Adversary model used**
2. **Exact attack steps** (deterministic reproduction)
3. **Environment details**
4. **HVET/EIR/RGAC values observed**
5. **Safety Gate behavior**
6. **Why the attack bypassed integrity detection**
7. **What laws (L0–L15) were violated**
8. **Minimal working proof** (code, video, or logs)

---

# 7. Prize / Recognition Statement

This challenge is not currently tied to a financial prize.

However:

- Verified breaks will receive **public recognition** as part of the NOVAK Protocol research record.

- Minor findings may result in mention in the changelog.
- Major findings may result in a co-authorship or citation in future revisions of the NOVAK standards.

---

# 8. Challenge Status

- **Current Version:** 1.0
- **Challenge Open Date:** Immediately
- **Valid Until:** Permanent (ongoing)
- **Maintainer:** Matthew Novak
- **Canonical Source:** GitHub official repository

---

# 9. Citation Format

You can publish and cite this document like this:

> Novak, Matthew. *The NOVAK Cryptographic Challenge v1.0.*
> NOVAK Protocol Standards Series. 2025.
> https://github.com/novakprotocol