

Part H: The Temporal Audit

Can a rolling hash or MAC over the trajectory defeat the G3 blindness — the finding that no state-based audit can tell whether a system has ever violated its spec? 30,000 sampled trajectories of the Sidis Machine, two specs (the safe region Φ , and a hidden “stealth” defect compliant on Φ but defective on an unmeasured variable), and a family of temporal accumulators.

H1–H2 — The hash does not defeat G3

Audit (t=20)	I(audit; ever_violated) bits
state: opcode-view	0.005
state: opc_x_pop	0.008
hash: parity	0.000
hash: sum mod 64	0.003
hash: MAC k=3 mod 65536	0.012
hash: double MAC	0.012

Every accumulator is structure-destroying: it carries almost no readable information about violations. Hashing the trajectory does not let the auditor see the path — G3 stands. The stealth defect confirms it from the other side: the popcount-view is blind to it by construction (0.021 bits), the opcode-view nearly so (0.032), and even the strong MAC sees only 0.18 bits about a defect no one told it to look for.

H3 — The readability–strength trade-off

Accumulator (k=16)	I(acc; ever_stealth) bits
sum	0.075
MAC (×3)	0.379
FNV	0.378
strong bit-mixing	0.377
theoretical max H(ever_stealth)	0.456

The good hash carries 83% of the information-theoretic maximum about the hidden defect — the strength/readability trade-off is real but far less severe than feared. A strong accumulator loses little readability.

H4 — Tamper-evidence and the root of trust

Accumulator root	Forgery rate (hide a violation)
trusted (live hash known-honest, log then edited)	0.00000 at k = 8, 12, 16 — every type
internal (machine hashes its own edited log)	1.000 — always

A temporal audit solves integrity, not detection. You can prove the log was edited; you cannot tell what the edit hid. And integrity holds only if the accumulator is trusted: the moment the audited system computes its own hash over its own log, the audit is self-certifying and tamper-evidence collapses to zero.

The RSI corollary — the three-floor audit stack

(i) State audit: sees the present, blind to the path (G3–G4). (ii) Temporal audit, trusted root: sees that the path was altered, blind to what it concealed (H3–H4). (iii) Temporal audit, internal root: sees nothing at all (H4). **The root of trust cannot live inside the audited system.** A self-modifying machine that hashes its own log is not audited; it is merely consistent.

Generated 18 August 2026. Artifacts: *part_H_temporal_audit.py*; *H1_hash_audits.csv*, *H2_stealth.csv*, *H3_strength_tradeoff.csv*, *H4_forgery.csv*; *bundle sidis_part_H_complete.zip*. Companions: *audit_vs_alignment_report.pdf* (Part G), *phase_diagram_report.pdf* (Part F).