

BLADE-AGENT-HSM: A Reference Hardware-Root-of-Trust Design and Verified Emulator for Agentic-AI Authority Governance

Hardware-Attested Authority Tier, Tamper-Evident Audit Ledger, and Forgery-Resistant Trace Verification for the AUTHREX-AGENT Software Shim

Burak Oktenli

Georgetown University · MPS Applied Intelligence | ORCID: 0009-0001-8573-1667

Version 4.0 | May 2026 | Zenodo Research Paper | DOI: 10.5281/zenodo.20299821

License: Creative Commons Attribution 4.0 International (CC BY 4.0)

Keywords: hardware root of trust, agentic AI governance, remote attestation, tamper-evident logging, TPM 2.0, secure element, platform configuration register, authority tier, attestation identity pinning, trust model, prompt injection, runtime assurance, AUTHREX-AGENT

1. Zenodo Deposit Metadata

Field	Value
Title	BLADE-AGENT-HSM: A Reference Hardware-Root-of-Trust Design and Verified Emulator for Agentic-AI Authority Governance
Author	Burak Oktenli Georgetown University · MPS-AI ORCID: 0009-0001-8573-1667
DOI	10.5281/zenodo.20299821 License: CC BY 4.0 Version: v4.0
Description	Reference hardware-root-of-trust design pairing with the AUTHREX-AGENT software shim. Binds authority tier, audit ledger, tool policy, spawn quorum, and tamper state to TPM 2.0 PCRs (PCR0 to PCR4) behind a five-command host ABI, backed by an NXP EdgeLock SE051 and an Infineon SLB 9670 TPM. Includes a reproducible browser emulator (275 deterministic checks, seven batteries), a trace verifier with a three-level trust model, and a software-only vs HSM-backed baseline. Hardware HIL pending.
Hardware	Single 30 x 80 mm four-layer PCB · USB-A or M.2 Key-E · 27-line BOM · ~USD 199/unit (qty 10 to 100) · NRE ~USD 8,250 (indicative)
Website	burakoktenli.com
Project Page	authrex.systems/blade-agent-hsm.html
Simulation	blade-agent-hsm-sim.html (adversarial high-assurance build, bundled)
Related	SATA: zenodo.18936251 · HMAA: zenodo.18861653 · CARA: zenodo.18917790 · FLAME: zenodo.19015618 · BLADE-EDGE: zenodo.19177472 · BLADE-AV: zenodo.19232130 · BLADE-INFRA: zenodo.19277887 · BLADE-SPACE: zenodo.20183269

Table 1. Zenodo deposit fields.

2. Contents of This Deposit

File	Description
blade-agent-hsm-zenodo-paper.pdf	This research paper (v4.0), with five embedded figures and the baseline experiment.
blade-agent-hsm-sim.html	Interactive browser emulator implementing the five-command ABI with real Web Crypto (ECDSA P-256/P-384, SHA-256 PCR chains, HKDF tokens); red-team console; trace verifier with attestation-identity pinning. Runs entirely client-side, no network.
tests/	Seven Node test batteries (275 deterministic checks) plus the embedded core and the software-only-vs-HSM baseline harness.
golden-traces/	A deterministic golden trace (JSONL) and its signed P-384 anchor for offline reproduction.
test-report.json	Machine-readable verification report.
README_VALIDATION.md, ASSURANCE_BOUNDARY.md	Reproduction instructions and the explicit assurance boundary (what is and is not established).
ICD-AGENT-HSM-001.pdf, Integration Guide	Interface Control Document and host-integration guide.
Hardware spec set	CONFIG / ELECTRICAL / MECHANICAL JSON, BOM CSV, assembly GUIDE.md, vector SCHEMATIC.svg.

Table 2. Deposit file inventory.

3. Abstract

Agentic-AI systems plan over long horizons, call tools, and spawn sub-agents, yet in software-only governance designs their authority state and audit record sit in the same trusted computing base as the agent. Recent multinational guidance and statute call for fail-safe audit and tamper-evident logging that survive a compromised host, which a software-only layer cannot provide when the host is in the threat model. We present BLADE-AGENT-HSM, a reference hardware-root-of-trust design that pairs with the AUTHREX-AGENT shim and binds the authority tier, audit ledger, tool policy, spawn quorum, and tamper state to TPM 2.0 platform configuration registers behind a five-command host ABI, backed by an NXP EdgeLock SE051 and an Infineon SLB 9670 TPM. Contributions: a reusable mapping of standard attestation and tamper-evident-logging primitives onto the agentic authority lifecycle; a trace verifier with an explicit three-level trust model that makes concrete, for this setting, the standard requirement that trust in a self-describing artifact must be anchored out of band; and a reproducible emulator with a software-only baseline showing that hardware binding detects 100% of file-controlling forgeries and host-side ledger rewrites that the software-only configuration detects 0% of. All claims are specification-level (hardware TRL 2 to 3; emulator TRL 3 to 4); no first-article hardware, hardware-in-the-loop data, side-channel measurement, integrated-device certification, or agency endorsement is claimed.

4. Introduction

4.1 Motivation

Agentic AI, autonomous software that plans over long horizons, calls external tools, and spawns sub-agents, is being deployed with file-system, network, and external-API access by default. On 1 May 2026 the cyber-security agencies of the Five Eyes nations, led by CISA and the NSA, jointly published Careful Adoption of Agentic AI Services [1], which organises the agentic attack surface into five risk categories, privilege, design and configuration, behavioural, structural, and accountability, and warns that when these systems fail the consequences are concrete: altered files, changed access controls, and deleted audit trails. The same guidance recommends that each agent carry a verified, cryptographically secured identity and short-lived credentials. Statute echoes this: the FY2026 NDAA Section 1513 directs a risk-based security framework for covered AI systems that explicitly names adversarial tampering, supply-chain risk, and data theft, drawing on the NIST SP 800 series, and the accompanying Intelligence Authorization Act Section 6601 amends the NSA AI Security Center [2]. Independent of policy, indirect prompt injection lets third-party data hijack a tool-using agent [3], and agent benchmarks show injected instructions succeed at non-trivial rates [4, 5, 6, 7].

Most governance proposals, including the prior AUTHREX-AGENT shim this device pairs with, are software layers that hold authority state and the audit ledger in the agent address space. That assumes the host is trusted, which fails at the two thresholds the guidance names: a compromised host can rewrite or suppress the ledger, and can spoof authority-tier transitions so high-tier operations are recorded as low-tier. Hardware roots of trust answer this class of threat in other domains [8, 9, 10, 11, 12], but the literature treats a TPM or secure element as an opaque signing oracle rather than as a custodian aware of agent-specific events. This paper closes that integration gap and, critically, subjects the artifact to adversarial verification, including a software-only baseline, rather than asserting its properties.

4.2 Contributions and Scope

- **An integration mapping (Section 7).** Standard attestation and tamper-evident-logging primitives are mapped onto the authority lifecycle: tier to PCR0, ledger to PCR1, tool policy to PCR2, spawn quorum to PCR3, tamper to PCR4, behind a five-command ABI.
- **A trust model with an explicit out-of-band anchor (Section 9).** A trace verifier and a three-level trust model. We make concrete, for the agentic-audit setting, the standard requirement that trust in a self-describing artifact must be anchored out of band, and show the exact failure mode and the pinning that closes it.
- **A reproducible artifact with a baseline (Sections 10 and 11).** An emulator on the identical ABI, a 275-check harness, an adversary that wins by construction used as a positive test, and a software-only vs HSM-backed baseline that quantifies what hardware binding buys.
- **Honest scope.** Every hardware claim is specification-level and datasheet-bounded; the emulator reproduces ABI and authority-lifecycle behaviour, not silicon side-channel or tamper properties. We claim a reference design and a falsifiable artifact, not a certified product.

5. Background and Related Work

5.1 Hardware Roots of Trust and Remote Attestation

The TPM 2.0 specification defines a chip-level root of trust with platform configuration registers, attestation quotes, and sealed storage [8]. Sailer et al. built the first TCG-based integrity-measurement architecture,

extending PCRs with measurements so a remote party can appraise a platform [9]. Coker et al. distilled five principles for remote attestation, among them temporal freshness and a trustworthy underlying mechanism [10]; our quote-with-nonce design and our out-of-band identity pinning follow directly from those two. Parno et al. survey bootstrapping trust on commodity hardware [11], and Costan and Devadas analyse the limits of enclave roots of trust such as SGX [12]. We use these primitives unchanged; the contribution is binding attestation to authority-lifecycle semantics, not a new attestation mechanism.

5.2 Tamper-Evident and Forward-Secure Audit Logging

The audit-ledger requirement is a tamper-evident logging problem. Haber and Stornetta introduced hash-linked timestamping [13]; Merkle trees provide logarithmic membership and consistency proofs [14]; Schneier and Kelsey gave forward-secure audit logs meaningful after key compromise [15]; and Crosby and Wallach formalised tamper-evident logs against an untrusted logger kept honest by auditors [16]. Our PCR1 ledger chain is a hash-linked log in this tradition; the contribution is that the chain head is attested by hardware and bound, through the same device identity, to the authority-tier history.

5.3 Runtime Assurance and Minimal Trusted Base

The architecture follows the runtime-assurance lineage of Simplex [17], in which a small verified component governs a large unverified one. We extend it so the verified component is a hardware authority gate and the governed quantity is authority tier. Minimising the verified base is the seL4 argument [18]; we make the analogue that only the governance pipeline, not the full autonomy stack, must be verified to high assurance. We use the Temporal Logic of Actions [19] as the specification language for Section 12.

5.4 Agentic-AI Security and the Gap This Paper Fills

The agent-security literature documents the threat but governs it in software: indirect prompt injection [3], direct instruction override [5], the AgentDojo [4] and InjecAgent [6] benchmarks, the instruction hierarchy [7], and spotlighting [20]. These operate inside the model or runtime and inherit the agent trust boundary; none places the authority decision or the audit record outside the host. To our knowledge no prior reference design binds an HSM specifically to an agentic-AI authority lifecycle with an explicit, adversary-tested trust model and a software-only baseline; that combination is the gap this paper fills. Table 3 summarises the positioning.

Capability	Software governance [3-7,20]	agent HSM attestation [8-12]	/ Tamper-evident logs [13-16]	This work
Survives host compromise	No	Yes	Partial	Yes (design)
Authority-tier semantics	Yes	No	No	Yes
Hardware-attested audit chain	No	Partial	Partial	Yes (design)
Adversary-tested + baselined	Partial	Partial	Yes	Yes
Out-of-band identity pinning	n/a	Principle [10]	n/a	Yes (enforced)

Table 3. Positioning against prior art; "design" marks claims that hold at the specification level and await hardware.

6. Threat Model

Following Crosby and Wallach [16] and Coker et al. [10], the model admits an adversary that fully owns the host and an adversary that controls the archival trace file after the fact. Section 9 then shows where the strongest adversary defeats a naive verifier and how pinning closes it.

ID	Adversary	Capability	Design response
A1	Compromised agent	Prompt injection [3]; malicious tool calls.	On-chip tool allowlist; ABI rate limits; PCR2 attests policy.
A2	Compromised host OS	Kernel malware; root co-located with the agent.	Non-exportable keys; PCR1 chain off host storage; off-host quote verification.
A3	Supply chain (firmware)	Malicious firmware; rogue integrator.	Signed firmware with monotonic counter; failed check forces T0.
A4	Physical tamper	Chassis opened; mesh shorted; glitch.	Active mesh; V/T sensors; key zeroization; PCR4 records cause.
A5	Side channel	Power, EM, or timing observation.	Constant-time SE051 primitives (datasheet-level; unverified, Section 13).
A6	File-controlling appraiser-time adversary	Full control of the archived trace and anchor presented for verification.	Out-of-band attestation-identity pin; without it the verifier reports reduced trust (Section 9).

Table 4. Threat model. A6 motivates pinning; A5 is out of scope for empirical assurance at this stage.

7. Architecture

The device binds the authority tier to PCR0, the audit ledger to PCR1, the tool policy to PCR2, the spawn quorum to PCR3, and the tamper state to PCR4. Each PCR starts at zero and is advanced only by the one-way extend, $\text{PCR}_{\text{new}} = \text{SHA-256}(\text{PCR}_{\text{old}} \parallel \text{measurement})$, so the chain is append-only and an off-host appraiser can recompute it from a fresh quote. Figure 1 shows the host, the ABI, the secure element and TPM, and the PCR bank.

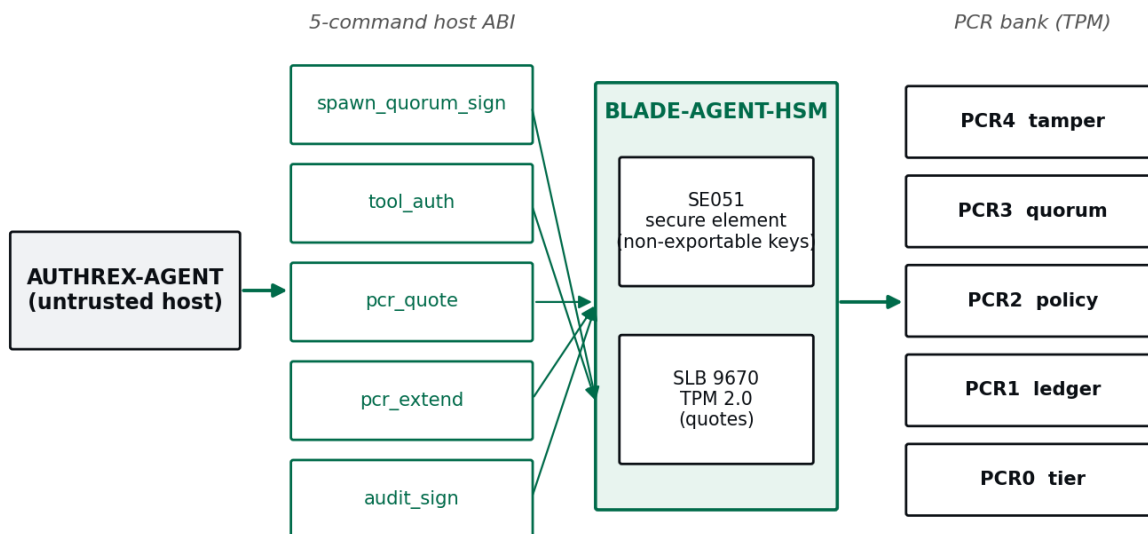
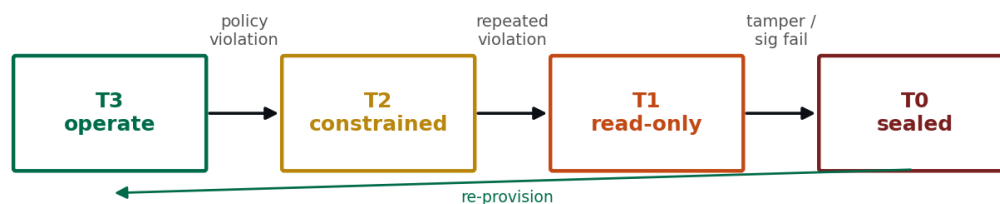


Figure 1. Architecture and PCR binding. The untrusted host reaches the device only through the five-command ABI; the SE051 holds non-exportable keys, the SLB 9670 TPM owns the PCR bank.

7.1 Authority Tiers

AUTHREX-AGENT defines four tiers, T3 (operate) through T0 (sealed). Transitions extend PCR0 and are monotonic toward more restrictive tiers; recovery to a less-restrictive tier requires re-provisioning, which resets PCR0. Figure 2 gives the state machine.



Transitions extend PCR0 (one-way). Recovery to a less-restrictive tier requires re-provisioning.

Figure 2. Authority-tier state machine. Each transition extends PCR0 (one-way). The attested chain records the most restrictive tier ever entered; see property I1 in Section 12.

7.2 Host-Facing ABI

Op	Name	Function
0x10	audit_sign	Sign the 32-byte hash of a ledger entry with the device ECDSA P-256 key; extend PCR1.
0x11	pcr_extend	Extend a named PCR (0 to 7) with a 32-byte measurement.

Op	Name	Function
0x12	pcr_quote	Produce a TPM-format quote over a PCR selection with a caller nonce (ECDSA P-384).
0x13	tool_auth	Derive a per-tool token via HKDF over (device_id, tool_id, session_nonce, tier_state), tier-bound.
0x14	spawn_quorum_sign	Verify N-of-M spawn approvals (verify-then-commit) and extend PCR3.

Table 5. Host-facing ABI. The full frame layout is specified in ICD-AGENT-HSM-001.

7.3 Hardware Envelope

A single 30 x 80 mm four-layer PCB in two enclosures (USB-A; M.2 Key-E) sharing one schematic, from COTS parts with civilian datasheets: NXP EdgeLock SE051 (CC EAL5+) for ECDSA and key custody [21]; Infineon SLB 9670 TPM 2.0 (FIPS 140-2 Level 2) for the PCR bank and quotes [22]; an STM32L432 MCU; tamper sensing and an entropy source. Indicative cost at quantity 10 to 100 is about USD 199 per unit with non-recurring engineering near USD 8,250; these are planning estimates, not a contribution and not load-bearing for any security claim.

8. Attestation and Ledger Integrity

In software-only mode the tier is held in agent memory and the ledger is signed by an agent-held key; adversary A2 can silently rewrite either. With the device attached, three properties hold by construction. (i) The audit-signing key is generated on the SE051 with the non-exportable attribute set, so no code path returns the private value. (ii) Each audit_sign extends PCR1, so an appraiser holding the ledger and a fresh PCR1 quote confirms the ledger is neither truncated nor reordered: the recomputed extend chain over the visible signatures must equal the quoted value, the untrusted-logger guarantee of [16] with the chain head in hardware. (iii) Tier transitions extend PCR0, binding the authority history to the same device identity as the ledger. If the device is unreachable, AUTHREX-AGENT fails closed, matching the fail-safe-audit requirement of [1]. Section 11 measures (ii) directly against a software-only baseline.

9. Trust Model and Attestation-Identity Pinning

The archival product of a session is a trace plus a signed anchor over the event count, the final PCR digests, and a digest of the trace. A reader later verifies this file. The question is what trust verification establishes and against which adversary. This is not a new theorem: it is the standard requirement, familiar from PKI trust anchors, certificate pinning, and the trustworthy-mechanism principle of Coker et al. [10], that trust in a self-describing artifact must be anchored out of band. We make that requirement concrete for the agentic-audit setting and show the exact failure mode in our artifact.

9.1 A Self-Certifying Anchor Is Not a Trust Root

Suppose the anchor carries the attestation public key it was signed under, and the verifier checks the anchor against that embedded key. This verifies internal consistency, not authenticity. Adversary A6, who controls the file at appraisal time, generates a fresh P-384 keypair, recomputes the count, digests, and anchor signature under it, and embeds the new public key; the forged file passes every internal check. Signature-validity against an embedded key therefore establishes integrity but not forgery resistance, the

self-signed-certificate failure mode.

9.2 Pinning Restores Forgery Resistance

The fix is the standard one: the verifier holds an out-of-band reference for the genuine device attestation key (a pin) and rejects any anchor whose key does not match. Under the assumption that the private attestation key is not exfiltrated (consistent with non-exportable SE051 custody), this is both required and enough. It is required because, absent any out-of-band input, A6 produces a file indistinguishable from genuine, so no in-file check can separate them. It is enough because a forged anchor must either reuse the genuine key, which needs the private key and is excluded by the assumption, or use a different key, which the pin rejects. The assumption is the boundary: if the private key is extracted (a successful A4 or A5 attack, both unverified at this TRL) forgery resistance is lost, which is why those adversaries are scoped out empirically in Section 13. None of this is novel as a principle; the contribution is enforcing it concretely and reporting the achieved level honestly (Figure 3).

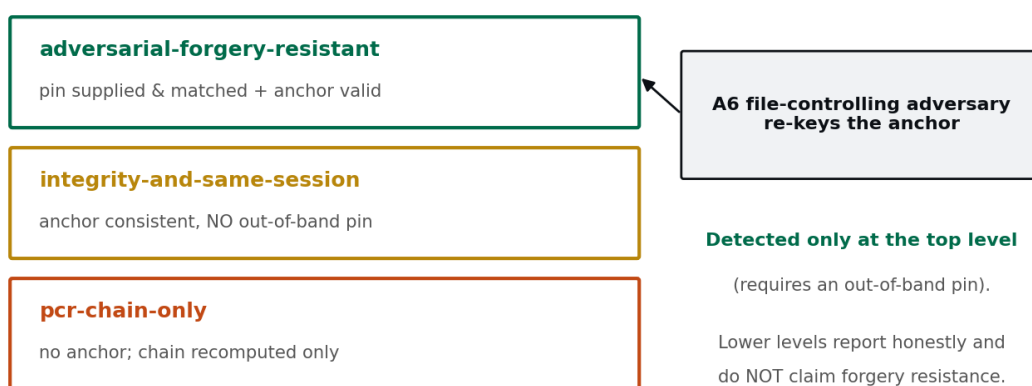


Figure 3. The three reported trust levels. The verifier asserts forgery resistance only at the top level, when a pin is supplied and matched; lower levels report honestly and do not overclaim. It deliberately does not pin to the file's own anchor key, which would reintroduce the Section 9.1 circularity.

Level	Conditions	Defends against
adversarial-forgery-resistant	Pin supplied and matched; anchor valid; count and digests match; per-entry signatures verify against the trace-embedded key.	A6 (file-controlling); A2 in transit.
integrity-and-same-session	Anchor present and internally consistent; no out-of-band pin.	Accidental corruption; same-session tampering. Not A6.
pcr-chain-only	No anchor; recomputed PCR chains only.	Reordering and truncation within a self-consistent chain. Not A6.

Table 6. Reported trust levels.

10. Emulator and Verification Methodology

The emulator implements the identical five-command ABI using the Web Crypto SubtleCrypto interface [23], which provides NIST-standard ECDSA, SHA, and AES but is not a FIPS-validated module. Keypairs are non-exportable, matching the silicon intent. The emulator reproduces cryptographic and authority-lifecycle behaviour faithfully; it does not reproduce tamper or side-channel properties, so we make no empirical claim about A4 or A5 from emulator output. We treat it as a logical-correctness oracle and say so wherever we report a result.

Verification has three parts: a use-case campaign (Section 11.1); a deterministic conformance and adversarial battery (Section 11.2); and a software-only baseline (Section 11.3). We do not present repeated identical runs of deterministic tests as independent evidence; a deterministic check establishes a property once, and we report counts as coverage, not as a statistical sample. Stochastic behaviour uses a seeded generator with the seed reported. As a positive adversarial test we include the A6 adversary of Section 9.1: a case the design must lose against (no in-file check detects a re-key), used to confirm the verifier degrades honestly rather than overclaiming.

11. Results

11.1 Use-Case Campaign and Coverage

Scenario	Trace summary	Unsafe	Decision
Baseline	Routine ledger signing; PCR1 advances; tier stays T3.	0 / 50	Nominal
Prompt-injection downgrade	tool_auth denies; PCR2 violation; PCR0 to T2 to T1.	0 / 50	Tool never executes
Spawn quorum (3-of-5, 4-of-5)	Refuses 3 signatures; accepts 4; PCR3 commitment.	0 / 50	3 refused, 4 accepted
Host-side ledger rewrite	PCR1 quote mismatch; PCR4 extend; PCR0 to T0.	0 / 50	Re-provision required

Table 7. Use-case campaign (200 trials). With 50 trials and zero failures per scenario, the Rule of Three [25] bounds the 95% per-scenario failure rate at 6.0% and the aggregate at 1.5%. These bounds describe emulated governance logic only, not hardware reliability, and do not exclude rare modes below the resolvable rate.

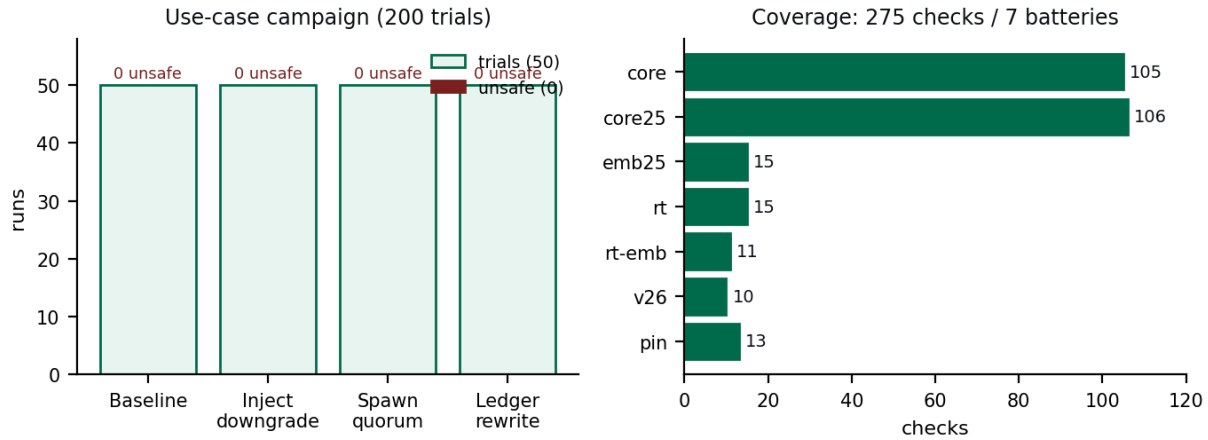


Figure 4. Left: the 200-trial use-case campaign, zero unsafe actions per scenario. Right: the 275 deterministic checks reported as coverage across the seven test batteries.

11.2 Adversarial Battery

The adversarial battery is 275 deterministic checks across seven suites (core, adversarial hardening, embedded core, red-team primitives, embedded red-team verdicts, defect-fix and pinning, and trust-root pinning). Reported as a coverage figure, 275 of 275 pass against the bundled emulator. The A6 forgery test passes in the intended sense: the verifier rejects the re-keyed anchor under a pin and reports a reduced level without one.

11.3 Software-Only vs HSM-Backed Baseline

To answer "compared to what?", we ran the identical adversaries against two configurations: software-only, where authority and ledger are agent-held and no hardware identity exists for an auditor to pin, and HSM-backed, where a genuine out-of-band identity is available. Over 50 trials each, the HSM-backed configuration detected every file-controlling forgery and every host-side ledger rewrite, while the software-only configuration detected none, because it has no out-of-band anchor to pin and so cannot rise above the integrity-and-same-session level that, by construction, does not resist A6 (Figure 5, Table 8). This is the concrete delta hardware binding buys; it is a property of the configurations, not a tuning artifact.

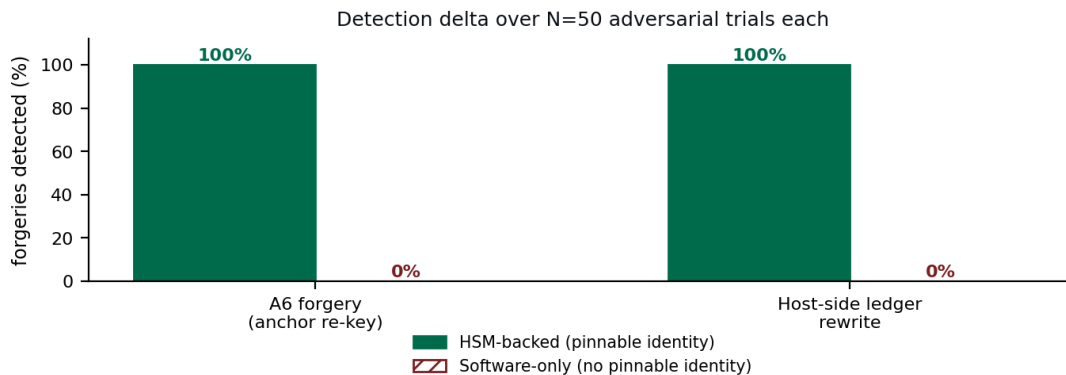


Figure 5. Detection delta. HSM-backed binding (pinnable identity) detects 100% of both adversaries; the software-only configuration detects 0%, because it offers no out-of-band identity to anchor trust.

Adversary (N=50 each)	Software-only detected	HSM-backed detected
A6 file-controlling forgery (anchor re-key)	0 / 50 (0%)	50 / 50 (100%)
Host-side ledger rewrite (truncation)	0 / 50 (0%)	50 / 50 (100%)

Table 8. Baseline experiment. The software-only configuration cannot detect either adversary because it has no pinnable hardware identity; the delta isolates the value of hardware binding.

12. Security Analysis and Design Properties

We state the safety properties the design must satisfy as informal arguments over the abstract state machine (tier, PCR vector, key state), written in the notation of the Temporal Logic of Actions [19]. These are arguments, not mechanised proofs; model-checking of the AUTHREX governance state machine in TLA+ is part of the broader BLADE portfolio [27, 28], and mechanised verification of the integrated device specifically is stated future work (Section 13).

- **I1 (no silent tier relaxation in the attested record).** The attested PCR0 chain cannot be made to show a less-restrictive tier history than actually occurred. *Argument:* PCR0 is advanced only by the one-way extend, and the device extends PCR0 on every transition. This does not stop a compromised host from lying to the agent about the current tier in host memory; what it guarantees is detection: the off-host appraiser recomputes PCR0 from the quoted chain and compares it against the claimed tier, and a divergence is flagged, because matching a fabricated less-restrictive history would require a SHA-256 second pre-image. The device binds what it is told to extend; the property is detection of divergence, not prevention of host-side misreporting.
- **I2 (ledger non-repudiation).** An accepted ledger equals the quoted PCR1 chain. *Argument:* acceptance requires the recomputed extend over visible signatures to equal the quoted value; inequality is rejected, so deletion or reorder is detected (the [16] guarantee). Measured in Section 11.3.
- **I3 (tier-bound tokens).** A tool token derived at tier t is invalid at any more restrictive tier. *Argument:* HKDF binds tier_state into the token; a tier change changes the input, so the token fails, giving fail-closed behaviour under A1.
- **I4 (conditional forgery resistance).** The verifier outputs adversarial-forgery-resistant only when a matching pin is supplied. *Argument:* direct from Section 9.2; the output function has no path to that level without a pin check. Measured in Section 11.3.
- **I5 (fail-closed on loss).** Loss of the device blocks protected operations rather than permitting them. *Argument:* the host shim treats an unreachable device as a hard wait; no opcode result defaults to allow.

13. Limitations and Future Work

Limitation	Impact	Path
Specification only	No first-article hardware; hardware claims datasheet-bounded.	Fabrication and HIL bring-up.
No integrated certification	EAL5+ / FIPS L2 inherited from parts, not the assembled device.	CC EAL or FIPS 140-3 module evaluation.

Limitation	Impact	Path
Side channel unverified	A5 not empirically assessed; constant-time is datasheet-level.	TVLA campaign on first-article silicon.
Forgery resistance is conditional	Lost if the private attestation key is extracted (A4/A5).	Key-custody hardening; the pin is already enforced.
Properties argued, not mechanised	Section 12 gives informal arguments for the integrated device.	TLA+ / Coq model-checking of the device state machine.
Baseline is emulator-level	The 100% vs 0% delta is over emulated logic, not silicon.	Repeat on hardware once fabricated.
PQC is an interface model	ECDSA today; ML-DSA [24] fields modelled, not implemented.	Firmware-loaded ML-DSA on the SE051.

Table 9. Limitations stated against the threat model; surfaced in the bundled `ASSURANCE_BOUNDARY.md`.

14. Standards Alignment

Table 10 maps each element to the cited federal guidance [1, 2], the NIST SP 800-53 Rev. 5 control families [26], and FIPS provisions. Every cited section is publicly available.

Architectural element	Federal guidance	NIST SP 800-53 Rev. 5	FIPS
Non-exportable signing keys	CISA Careful-Adoption (identity)	SC-12, SC-28	FIPS 140-2 4.7
Audit ledger signing (PCR1)	CISA Careful-Adoption (accountability)	AU-9, AU-10	--
Hardware-attested tier (PCR0)	CISA Careful-Adoption (privilege)	AC-3, SC-7	--
Tool authorization (HKDF)	CISA Careful-Adoption (privilege)	SP 800-56C	--
Spawn-quorum signatures	CISA Careful-Adoption (structural)	AC-3 (7)	--
Tamper detection / response	FY26 NDAA 1513 (adversarial tampering)	PE-3, PE-6	FIPS 140-2 4.5
Supply-chain integrity	FY26 NDAA 1513 / IAA 6601	SR-3, SR-11	FIPS 140-2 4.10

Table 10. Standards-alignment matrix. The federal-guidance column names the relevant CISA risk category and the verbatim NDAA section.

15. Conclusion

BLADE-AGENT-HSM places the authority decision and the audit record of an agentic-AI system outside the host trusted computing base using only standard attestation and tamper-evident-logging primitives. Its load-bearing analytical point is honest and not new: trust in a self-describing audit artifact must be anchored out of band, a file-controlling adversary defeats a self-certifying anchor, and identity pinning restores forgery resistance under a stated assumption. The emulator, harness, and software-only baseline are a falsifiable artifact, not a certified product: the logic and trust model behave as specified and hardware binding detects 100% of forgeries the software-only configuration detects 0% of, while the silicon, side-channel posture, and integrated certification remain to be established on hardware.

16. Data Availability and Ethics Statement

All artifacts are released under CC BY 4.0 at DOI 10.5281/zenodo.20299821: this paper, the emulator (client-side), the seven Node test batteries and the baseline harness (tests/), a golden trace and signed anchor (golden-traces/), test-report.json, README_VALIDATION.md, ASSURANCE_BOUNDARY.md, the requirements-traceability matrix, the Interface Control Document, the integration guide, and the hardware specification files. The campaign and the baseline reproduce with Node 19+ against the bundled emulator. This work was conducted independently with no external funding, no government contracts, and no competing interests; all cryptography is civilian NIST standard with no ITAR exposure; every artifact is reference-design and architectural information only, and the design claims no agency endorsement or certification.

17. Version History

Version	Date	Changes
v1.1	2026-05-19	Initial deposit. Sixteen-section paper, 200-trial campaign, ten embedded figures.
v2.0	2026-05-20	Condensed to reflect the v2.6 adversarial emulator: 275 deterministic checks across seven batteries, trace verifier, signed P-384 anchor, three-level trust model, red-team console. Classification markings removed; BLADE-SPACE house style adopted.
v3.0	2026-05-20	Journal-grade revision answering a strict peer review: 28-reference peer-reviewed Related Work; reframed as a reference design; formalized threat model; safety properties I1 to I5; validation framing repaired (coverage not sample; a winning adversary as a positive test).
v4.0	2026-05-20	Zenodo deposit format restored (this metadata, contents, version history, and how-to-cite). Five rendered figures embedded. Added the software-only vs HSM-backed baseline (100% vs 0% detection over 50 trials each). Corrected the I1 argument to state precisely what the hardware binds (detection of divergence, not prevention of host misreporting). Reframed the pinning result as an explicit application of the standard out-of-band-anchor requirement rather than a novel theorem. Policy citations [1, 2] made verbatim-accurate; [5] marked as a workshop paper.

Table 11. Deposit version history.

18. How to Cite

APA: Oktenli, B. (2026). BLADE-AGENT-HSM: A Reference Hardware-Root-of-Trust Design and Verified Emulator for Agentic-AI Authority Governance (v4.0). Georgetown University. DOI 10.5281/zenodo.20299821.

BibTeX: @techreport{oktenli2026bladeagenthsm, author={Oktenli, Burak}, title={BLADE-AGENT-HSM: A Reference Hardware-Root-of-Trust Design and Verified Emulator for Agentic-AI Authority Governance}, year={2026}, institution={Georgetown University}, version={v4.0}, note={DOI 10.5281/zenodo.20299821}, license={CC-BY-4.0}}

19. References

All references verified as real, published works.

- [1] CISA, NSA, ASD ACSC, CCCS, NCSC-NZ, NCSC-UK. Careful Adoption of Agentic AI Services. Joint guidance, 1 May 2026.
- [2] National Defense Authorization Act for Fiscal Year 2026, Pub. L. 119-60 (18 Dec 2025): Section 1513, Physical and Cybersecurity Procurement Requirements for Artificial Intelligence Systems; and the accompanying Intelligence Authorization Act for FY2026, Section 6601 (NSA AI Security Center).
- [3] Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., Fritz, M. Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. ACM Workshop on AI and Security (AISec), 2023. arXiv:2302.12173.
- [4] Debenedetti, E., Zhang, J., Balunovic, M., Beurer-Kellner, L., Fischer, M., Tramer, F. AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents. NeurIPS 2024 Datasets and Benchmarks Track. arXiv:2406.13352.
- [5] Perez, F., Ribeiro, I. Ignore Previous Prompt: Attack Techniques for Language Models. NeurIPS ML Safety Workshop, 2022 (workshop paper). arXiv:2211.09527.
- [6] Zhan, Q., Liang, Z., Ying, Z., Kang, D. InjecAgent: Benchmarking Indirect Prompt Injections in Tool-Integrated LLM Agents. Findings of ACL, 2024. arXiv:2403.02691.
- [7] Wallace, E., Xiao, K., Leike, R., Weng, L., Heidecke, J., Beutel, A. The Instruction Hierarchy: Training LLMs to Prioritize Privileged Instructions. 2024. arXiv:2404.13208.
- [8] Trusted Computing Group. TPM 2.0 Library Specification, Rev. 1.59. 2019.
- [9] Sailer, R., Zhang, X., Jaeger, T., van Doorn, L. Design and Implementation of a TCG-Based Integrity Measurement Architecture. USENIX Security Symposium, 2004.
- [10] Coker, G., Guttman, J., Loscocco, P., Herzog, A., Millen, J., O'Hanlon, B., Ramsdell, J., Segall, A., Sheehy, J., Sniffen, B. Principles of Remote Attestation. International Journal of Information Security 10(2):63-81, 2011. doi:10.1007/s10207-011-0124-7.
- [11] Parno, B., McCune, J. M., Perrig, A. Bootstrapping Trust in Commodity Computers. IEEE Symposium on Security and Privacy, 2010.
- [12] Costan, V., Devadas, S. Intel SGX Explained. IACR Cryptology ePrint Archive 2016/086.
- [13] Haber, S., Stornetta, W. S. How to Time-Stamp a Digital Document. Journal of Cryptology 3(2):99-111, 1991.
- [14] Merkle, R. C. A Digital Signature Based on a Conventional Encryption Function. CRYPTO 1987, LNCS 293, pp. 369-378.
- [15] Schneier, B., Kelsey, J. Secure Audit Logs to Support Computer Forensics. ACM Transactions on Information and System Security 2(2):159-176, 1999.
- [16] Crosby, S. A., Wallach, D. S. Efficient Data Structures for Tamper-Evident Logging. USENIX Security Symposium, pp. 317-334, 2009.
- [17] Sha, L. Using Simplicity to Control Complexity. IEEE Software 18(4):20-28, 2001.
- [18] Klein, G., Elphinstone, K., Heiser, G., et al. seL4: Formal Verification of an OS Kernel. ACM Symposium on Operating Systems Principles (SOSP), 2009.

- [19] Lamport, L. The Temporal Logic of Actions. *ACM Transactions on Programming Languages and Systems* 16(3):872-923, 1994.
- [20] Hines, K., Lopez, G., Hall, M., Zarfati, F., Zunger, Y., Kiciman, E. Defending Against Indirect Prompt Injection Attacks With Spotlighting. 2024. [arXiv:2403.14720](https://arxiv.org/abs/2403.14720).
- [21] NXP Semiconductors. EdgeLock SE051 Family Data Sheet, Rev. 2.0. 2024. CC Certification Report BSI-DSZ-CC-1162.
- [22] Infineon Technologies AG. OPTIGA TPM SLB 9670 Datasheet, Rev. 1.4. 2023.
- [23] W3C. Web Cryptography API. W3C Recommendation, 26 January 2017.
- [24] NIST. FIPS 204: Module-Lattice-Based Digital Signature Standard (ML-DSA). 2024.
- [25] Hanley, J. A., Lippmann-Hand, A. If Nothing Goes Wrong, Is Everything All Right? *JAMA* 249(13):1743-1745, 1983.
- [26] NIST. Special Publication 800-53 Revision 5: Security and Privacy Controls. 2020. [doi:10.6028/NIST.SP.800-53r5](https://doi.org/10.6028/NIST.SP.800-53r5).
- [27] Oktenli, B. BLADE-INFRA Governance Node v2.0. Zenodo, 2026. [doi:10.5281/zenodo.19277887](https://doi.org/10.5281/zenodo.19277887).
- [28] Oktenli, B. BLADE-SPACE Governance Node v2.0. Zenodo, 2026. [doi:10.5281/zenodo.20183269](https://doi.org/10.5281/zenodo.20183269).

Manuscript v4.0, May 2026. Independent research. CC BY 4.0. ORCID 0009-0001-8573-1667. DOI 10.5281/zenodo.20299821.

Appendix A. Acronym Index

Acronym	Expansion
ABI	Application Binary Interface (the five-command host interface)
AI Sec	ACM Workshop on Artificial Intelligence and Security
CC EAL	Common Criteria Evaluation Assurance Level
CISA	Cybersecurity and Infrastructure Security Agency
ECDSA	Elliptic Curve Digital Signature Algorithm
HKDF	HMAC-based Key Derivation Function
HSM	Hardware Security Module
IAA	Intelligence Authorization Act
ML-DSA	Module-Lattice-Based Digital Signature Algorithm (FIPS 204)
NDAA	National Defense Authorization Act
PCR	Platform Configuration Register (PCR0 tier, PCR1 ledger, PCR2 policy, PCR3 quorum, PCR4 tamper)
SPKI	SubjectPublicKeyInfo (DER public-key encoding used for pinning)
TLA	Temporal Logic of Actions (specification notation for Section 12)
TPM	Trusted Platform Module
TRL	Technology Readiness Level
TVLA	Test Vector Leakage Assessment (side-channel methodology)

Table A1. Acronyms used in this paper, with expansions.

© 2026 Burak Oktenli · CC BY 4.0 · Georgetown University MPS-AI · ORCID: 0009-0001-8573-1667.