SATA Sensor Trust Simulation

Simulation User Guide

Version: v3.8.9
Document: SATA-SIM-UG-001
Date: April 2026
Revision: 1.0

Author: Burak Oktenli
Institution: Georgetown University, School of Continuing Studies
Program: M.P.S. Applied Intelligence (STEM)
ORCID: 0009-0001-8573-1667

Related Artifact: DOI: 10.5281/zenodo.18936251

Document Control

Document ID	SATA-SIM-UG-001
Simulation Version	v3.8.9
Author	Burak Oktenli — Georgetown University — ORCID 0009-0001-8573-1667
Date / Revision	April 2026 / Rev 1.0 — Initial Release
Classification	UNCONTROLLED — Research Artifact
Related Artifact	DOI: 10.5281/zenodo.18936251

Table of Contents

1. Purpose, Scope, and Assumptions

This guide provides operating procedures for the SATA (Sensor Attestation and Trust Anchoring) simulation. SATA implements weighted Dempster-Shafer sensor trust fusion across multiple sensor channels with cross-sensor validation, temporal decay, TPM-anchored hardware attestation, and attack detection with confusion matrix analysis. Eight structured simulation sequences (SIM-01 through SIM-08) cover nominal operations, cryptographic attacks, sensor degradation, and full attack-recovery arcs.

Intended Audience: EB2-NIW petition evaluators, defense/aerospace reviewers, academic peers, and technical collaborators seeking independent verification of governance pipeline behavior.

Scope: Operation of the SATA Sensor Trust Simulation simulation. Does not cover mathematical theory (see published paper) or hardware specifications (see Blueprint.am).

1.1 Assumptions and Constraints

- The user has a modern browser with JavaScript enabled and HTTPS access.
- All parameter values are synthetic research placeholders, not calibrated against physical hardware.
- The simulation models governance logic only. Physical dynamics (aerodynamics, acoustics, electromagnetics) are simplified.
- Cryptographic operations (SHA-256 audit chain) use the WebCrypto API, not hardware TPM/HSM.
- Results are valid for demonstrating architectural behavior, not for operational safety assessment.

> **IMPORTANT**
> This simulation is a research prototype. Not for operational planning, safety-critical decisions, or system certification. All parameters are synthetic.

2. Quick Start

Step 1. Open sata-simulation.html in Chrome over HTTPS.

Step 2. Click AUTO DEMO to run all 8 structured simulations (SIM-01 through SIM-08) automatically.

Step 3. Or click RUN and manually inject sensor faults to test trust fusion.

Step 4. Monitor the confusion matrix for detection accuracy (TP/FP/TN/FN).

Step 5. Click VERIFY to check hash chain integrity of the audit ledger.

> **NOTE**
> All computation runs client-side. No data leaves your browser. Requires HTTPS (not file://).

3. System Requirements and Security Considerations

Browser	Chrome 90+, Firefox 88+, Safari 15+, Edge 90+

Protocol	HTTPS required (WebCrypto API for SHA-256 audit hashing)
Display	Min 1280x720; recommended 1920x1080+
CPU/Memory	Any modern processor. Monte Carlo (100+ runs): multi-core recommended, ~200MB RAM peak
GPU	WebGL-capable recommended for 3D visualizations (Three.js)
Network	Internet for initial CDN load (~500KB). All computation client-side after load.
Installation	None — zero install, no login, no backend, no database, no cookies

3.1 Security Considerations

• **No data exfiltration:** All computation runs in the browser. No data is sent to any server.

• **CDN dependencies:** React, ReactDOM, and Babel load from cdnjs.cloudflare.com (Cloudflare CDN with SRI hashes where available).

• **Audit integrity:** SHA-256 hash chain via WebCrypto API. Each audit entry links to the previous entry's hash. VERIFY button recomputes the entire chain.

• **No authentication:** The simulation has no login system. All state is ephemeral in browser memory.

4. Interface Layout and Navigation

4.1 Panel Layout

The interface uses a three-panel layout. **Left Panel:** Sensor health matrix showing per-sensor trust scores (tau) with color-coded status indicators (green=normal, yellow=warning, red=alarm). Fault injection controls below. **Center Panel:** Large authority gauge with weighted Dempster-Shafer fusion score, constellation heatmap showing cross-sensor agreement patterns, and trust timeline chart. **Right Panel:** Hash-chained attestation ledger, confusion matrix (TP/FP/TN/FN), and structured simulation controls (SIM-01 through SIM-08). **Top Bar:** RUN, STEP, RESET, AUTO DEMO, VERIFY, REPORT, DOWNLOAD, RUN MC buttons.

4.3 Panel Descriptions

Sensor Health Matrix. Real-time health status and trust score (tau) for each sensor channel

Constellation Heatmap. Visual representation of cross-sensor agreement and disagreement patterns

Authority Gauge. Resulting authority level from weighted Dempster-Shafer trust fusion

Confusion Matrix. Cumulative detection statistics: True Positive, False Positive, True Negative, False Negative

Trust Timeline. Historical trust scores over time for all sensor channels

> **TIP**
> Hover over interface elements for tooltips. Most gauges include ARIA labels for screen reader accessibility.

5. Operating Procedures

5.1 Startup

1. Navigate to the simulation URL or click Launch Simulation from burakoktenli.com.

2. Wait for loading (2-5 seconds). CDN scripts load from cdnjs.cloudflare.com.

3. Verify interface loads completely. All panels should be visible.

5.2 Standard Operation

1. Open the simulation via Launch Simulation or navigate to sata-simulation.html.

2. Click RUN to start the simulation tick loop. The sensor health matrix will show live trust scores.

3. Click AUTO DEMO to run all 8 structured simulation sequences (SIM-01 through SIM-08) automatically.

4. Alternatively, select a specific structured simulation (SIM-01 through SIM-08) from the scenario panel.

5. To manually inject faults, use the fault injection controls: select a sensor channel and fault type (stuck, drift, spike, noise).

6. Monitor the confusion matrix as attacks are detected or missed. TP/FP/TN/FN counters update in real time.

7. Click VERIFY to check the hash chain integrity of the audit ledger.

8. Click REPORT to generate a structured verification report.

9. Click RUN MC to execute 100 Monte Carlo trials with statistical analysis.

10. Use DOWNLOAD to export session data as JSON for independent verification.

5.3 Shutdown

1. Export session data. **2.** Close browser tab (all state discarded).

> **IMPORTANT**
> State is not persisted. Export before closing to preserve results.

6. Parameter Reference

Parameters are pre-configured per scenario. See Section 4 for interface controls and the published paper for parameter derivation.

7. Scenario Reference

SIM-01	Nominal Healthy	All sensors normal, chain grows cleanly	tau >= 0.95, all green
SIM-02	Signature Forgery	AIK signature forgery attack	Blocked immediately, tau drops to 0.70
SIM-03	Counter Rollback	Replay with old TPM counter	Counter mismatch detected
SIM-04	PCR State Injection	Compromised boot state hash	PCR verification fails
SIM-05	Stale Nonce / Replay	Nonce age check attack	Age check fires, attestation rejected

SIM-06	Sensor Degradation	Progressive quality loss	tau decays smoothly, tier drops gracefully
SIM-07	Multi-Vector Attack	SIG + CTR simultaneously	tau degrades, non-zero if 1 sensor clean
SIM-08	Attack -> Recovery	Full CARA cycle	Lockout -> GREP I-IV -> re-entry at T2

8. Metrics, Formulas, and Verification

8.1 Key Metrics

Per-Sensor Trust (tau_i)

Individual trust score for each sensor channel, computed from attestation verification, temporal decay, and cross-sensor validation.

Aggregate Trust

Weighted Dempster-Shafer fusion of all sensor channels. Weights reflect sensor reliability and operational importance.

Cross-Sensor Validation

Pairwise agreement scores between sensor channels. Disagreement triggers trust penalties on the divergent sensor.

Detection Performance

Confusion matrix (TP, FP, TN, FN) tracking attack detection accuracy across the session.

Hash Chain Integrity

SHA-256 linked audit entries. VERIFY button recomputes the entire chain to detect any tampering.

8.2 Verification Checklist

Perform the following checks to verify correct simulation behavior:

Run SIM-01 (Nominal)	tau >= 0.95, all sensors green.
Run SIM-02 (Signature Forgery)	Attack blocked. tau drops to ~0.70.
Run SIM-07 (Multi-Vector)	tau degrades. Non-zero if 1 sensor clean.
Run SIM-08 (Attack -> Recovery)	Full GREP cycle: lockout -> Phase I-IV -> re-entry.
Click VERIFY	Hash chain integrity confirmed.
Check confusion matrix	TP/FP/TN/FN counters updated after attacks.

9. Data Export and Reproducibility

Click export/download to save session JSON with parameters, history, and audit trail.

Verification: 1) Export JSON. 2) Note PRNG seed. 3) Reload with same seed/params. 4) Verify bit-exact match.

9.1 Reproducibility Guarantee

PRNG	Mulberry32 (32-bit seeded)
Math.random()	Zero calls in computation paths
Cross-Browser	Verified: Chrome, Firefox, Safari, Edge
Cross-Platform	Verified: Windows, macOS, Linux
Audit Chain	SHA-256 via WebCrypto (SubtleCrypto API)

10. Limitations and Threat Considerations

Simulation-Only Evidence	Browser-based. No physical sensor data or real TPM hardware.
Simulated TPM	AIK signatures, PCR extends, and nonce challenges are modeled, not hardware-backed.
Fixed Sensor Weights	Dempster-Shafer weights are pre-configured, not learned from operational data.
No Environmental Noise	Sensor noise uses Gaussian models, not physics-based sensor simulation.
Binary Fault Model	Faults are injected discretely (stuck/drift/spike). Gradual degradation is simplified.

10.1 Threat Considerations

• **CDN compromise:** React/Babel load from cdnjs.cloudflare.com. A CDN compromise could inject malicious code. Mitigation: Subresource Integrity (SRI) hashes on script tags where available.

• **Browser extensions:** Malicious extensions could modify simulation DOM/state. Mitigation: test in Incognito mode for clean results.

• **Local modification:** Users can modify simulation code via DevTools. Exported data should be verified against the published source on burakoktenli.com.

11. Troubleshooting

Trust scores stuck at 1.0	No faults injected	Inject a sensor fault or run a structured simulation (SIM-02+).
Confusion matrix all zeros	No attacks detected yet	Run SIM-02 (Signature Forgery) or SIM-07 (Multi-Vector) first.
Hash chain verification fails	Simulated chain corruption scenario	This may be intentional in SIM scenarios. Click RESET and re-verify.
AUTO DEMO stops mid-sequence	Browser tab inactive	Keep the browser tab active and focused during demo.

Sensor health matrix not updating	Simulation paused	Click RUN to resume or STEP to advance one tick.

12. Glossary and References

12.1 Glossary

SATA	Sensor Attestation and Trust Anchoring — weighted Dempster-Shafer trust fusion
tau (trust)	Per-sensor trust score computed from attestation, temporal decay, and cross-validation
Dempster-Shafer	Evidence theory for combining sensor trust measurements from multiple sources
TPM	Trusted Platform Module — hardware root of trust for attestation (simulated)
AIK	Attestation Identity Key — cryptographic key for TPM attestation signatures
PCR	Platform Configuration Register — TPM register storing boot state measurements
Confusion Matrix	Detection statistics: True Positive, False Positive, True Negative, False Negative
Cross-Validation	Pairwise comparison between sensor channels to detect inconsistencies

12.2 References

[1] Oktenli, B. (2026). SATA: A Hardware-Anchored tau-Chain Protocol. Zenodo. DOI: 10.5281/zenodo.18936251.

[2] Oktenli, B. (2026). SATA Simulation v3.8.9. https://burakoktenli.com/sata-simulation.html

[3] Shafer, G. (1976). A Mathematical Theory of Evidence. Princeton University Press.

[4] Trusted Computing Group (2019). TPM 2.0 Library Specification.

[5] U.S. DoD (2023). Directive 3000.09: Autonomy in Weapon Systems.

12.3 Contact

Burak Oktenli — Georgetown University, School of Continuing Studies
Website: **burakoktenli.com** | ORCID: **0009-0001-8573-1667**
For questions about this simulation or the governance architecture research program, use the contact form at burakoktenli.com.

End of Document