# ADARA Deception Detection Simulation

## Simulation User Guide

Version: v10.0
Document: ADARA-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.19043924

# Document Control

| | |
|---|---|
| **Document ID** | ADARA-SIM-UG-001 |
| **Simulation Version** | v10.0 |
| **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.19043924 |

## Table of Contents



 

# 1. Purpose, Scope, and Assumptions

This guide provides operating procedures for the ADARA (Adversarial Deception-Aware Risk Architecture) simulation. ADARA implements a proactive Deception Probability Engine (DPE) that computes P(adversarial) from input distribution anomalies, temporal correlation, cross-sensor consistency, and Bayesian mission history. Authority is adjusted pre-emptively: A_adj = A_hmaa x (1 - lambda x P_deception). The simulation includes 7 analysis tabs, operator labeling for ROC curve construction, Red Team mode for manual adversary control, encrypted session export, and full tick-level replay.

**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 ADARA Deception Detection 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 adara-simulation.html in Chrome over HTTPS.

**Step 2.** Select a scenario (NOMINAL, GPS_SPOOFING, ML_ADVERSARIAL, COMPOUND, ASYMMETRIC).

**Step 3.** Click Run to start. Monitor P(deception) rising under attack scenarios.

**Step 4.** Label alerts as TRUE or FALSE to build the ROC curve on the Metrics tab.

**Step 5.** Switch to Red Team tab to manually control adversarial sensor inputs.

> **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 full-width React application layout. **Header Bar:** Real-time threat level indicator (LOW/ELEVATED/HIGH/CRITICAL with color), P(deception) value, adjusted authority A_adj, OODA-loop alert feed with aging timers, and role selector (Commander/Operator). **Tab Bar:** Seven tabs — Dashboard, Sensors, Adversary, Audit, Metrics, Red Team, Replay. **Dashboard Panel:** DPE gauges per sensor, authority gauge, scenario selector, lambda/authority sliders, run controls. **Footer:** Version, copyright, SHA-256 chain status, legal links.

### 4.2 Navigation Tabs

| | |
|---|---|
| Dashboard | Main view: threat level, DPE gauges, authority gauge, sensor panel, alert feed |
| Sensors | Per-sensor deception probability scores with confidence intervals |
| Adversary | Adversary model parameters: asymmetric grid, correlation matrix, attack signatures |
| Audit | SHA-256 hash-chained audit trail with verification button |
| Metrics | ROC curve, FAR/FRR analysis, detection performance statistics |
| Red Team | Manual adversary control: set individual sensor values to test detection |
| Replay | Tick-by-tick replay with full state reconstruction and threat/authority overlays |

### 4.3 Panel Descriptions

**Threat Level Indicator.** Current classification: LOW (green), ELEVATED (yellow), HIGH (orange), CRITICAL (red)

**DPE Gauges.** Per-sensor deception probability estimates with confidence intervals

**Authority Gauge.** A_adj = A_hmaa x (1 - lambda x P_d) showing deception-adjusted authority

**Alert Feed.** Real-time alerts with OODA-loop timing (warning at configurable seconds, critical at 2x)

**ROC Curve.** Receiver Operating Characteristic built from operator-labeled alerts

> **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 adara-simulation.html.

**2.** Select a scenario from the dropdown (NOMINAL, GPS_SPOOFING, ML_ADVERSARIAL, COMPOUND, ASYMMETRIC).

**3.** Click the Run button to start the simulation tick loop.

**4.** Monitor the Dashboard: observe P(deception) rising when attack scenarios inject adversarial inputs.

**5.** When alerts appear, click to label them TRUE (confirmed deception) or FALSE (false alarm). Labels build the ROC curve on the Metrics tab.

**6.** Adjust the Lambda slider (0.00-1.00) to control deception sensitivity. Higher lambda = more aggressive authority reduction.

**7.** Switch to Red Team tab to manually control individual sensor inputs and test detection boundaries.

**8.** Switch to the Audit tab and click VERIFY to check hash chain integrity across all recorded entries.

**9.** Use Replay tab to scrub through past ticks. Inspect any historical moment's full state.

**10.** Click EXPORT to download session data as JSON. Use the encryption option with a passphrase for .adara encrypted exports.

## 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

| | | | | |
|---|---|---|---|---|
| Lambda | Slider | 0.00 - 1.00 | 0.85 | Deception sensitivity: scales authority reduction |
| A_hmaa | Slider | 0.00 - 1.00 | 0.78 | Base HMAA authority before deception adjustment |
| Prior P(d) | Slider | 0.00 - 1.00 | 0.05 | Bayesian prior probability of deception |
| A Floor | Slider | 0.00 - 1.00 | 0.00 | Minimum permissible adjusted authority |
| A Ceiling | Slider | 0.00 - 1.00 | 1.00 | Maximum permissible adjusted authority |

**NOTE**

All defaults are synthetic. Replace with empirically derived values before operational use.

# 7. Scenario Reference

| | | | |
|---|---|---|---|
| NOM | Nominal | Normal operations, no adversary | P(d) near prior, A_adj stable |
| GPS | GPS Spoofing | Coordinated GPS manipulation | P(d) rises, authority drops |
| ML | ML Adversarial | Adversarial ML perturbations | Subtle distribution shift detected |
| CMP | Compound Attack | Multi-vector simultaneous | Multiple DPE channels spike |
| ASY | Asymmetric | Unequal sensor weights exploited | Weighted channels show higher P(d) |

# 8. Metrics, Formulas, and Verification

## 8.1 Key Metrics

### P(deception)

Aggregate probability that current sensor inputs are adversarially manipulated. Computed from four DPE stages: input distribution anomaly, temporal correlation, cross-sensor consistency, and Bayesian prior update.

### A_adj (Adjusted Authority)

A_adj = A_hmaa x (1 - lambda x P_d). Pre-emptive authority reduction based on deception probability. Lambda controls sensitivity.

### Threat Classification

LOW (P_d < 0.15), ELEVATED (0.15-0.40), HIGH (0.40-0.70), CRITICAL (> 0.70). Each level triggers escalating governance responses.

### ROC Curve

Constructed from operator-labeled alerts. Shows True Positive Rate vs. False Positive Rate at varying P(d) thresholds.

### FAR / FRR

False Alarm Rate and False Rejection Rate computed from labeled alerts. Target: FAR < 5% for operational deployment.

#### SHA-256 Audit Chain

Every state transition hashed with SHA-256 (WebCrypto) and linked to the previous entry. Provides tamper-evident record of all DPE computations.

### 8.2 Verification Checklist

Perform the following checks to verify correct simulation behavior:

| | |
|---|---|
| Start simulation (RUN/START) | Interface loads. Governance pipeline begins processing. |
| Observe default state | Authority at nominal level. All pipeline stages PASS. |
| Inject a fault or attack | Authority reduces proportionally. Affected stage shows FAIL. |
| Monitor recovery | If CARA active, observe GREP recovery phases. |
| Export session data (JSON) | File downloads with parameters, history, and audit trail. |
| Reload and verify reproducibility | Same seed + params = identical outputs. |

## 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 computation. No physical sensor data or hardware measurements. |
| Uncalibrated Parameters | All values are synthetic research parameters, not empirically derived. |
| No Real-Time Guarantees | JavaScript engine provides no timing guarantees for safety-critical operations. |
| Simulated Cryptography | SHA-256 uses WebCrypto. TPM/HSM operations are modeled, not hardware-backed. |

| | |
|---|---|
| Single-Session State | All state held in memory. Closing the tab discards all data. |

### 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

| | | |
|---|---|---|
| Black screen after loading | React render error or CSP violation | Open F12 Console for error details. Try Chrome Incognito mode. |
| Simulation runs slowly | CPU-intensive Monte Carlo or 3D rendering | Close other browser tabs. Reduce sample count. |
| Controls not responding | Browser tab lost focus | Click inside the simulation window. Ensure tab is active. |
| Export button not working | Pop-up/download blocked | Allow downloads from the simulation domain in browser settings. |
| Loading screen never completes | CDN scripts blocked by firewall/extension | Disable ad blockers. Allow cdnjs.cloudflare.com. |

## 12. Glossary and References

### 12.1 Glossary

| | |
|---|---|
| ADARA Deception Detection | ADARA (Adversarial Deception-Aware Risk Architecture) |
| SATA | Sensor Attestation and Trust Anchoring — trust fusion |
| HMAA | Human-Machine Authority Architecture — authority computation |
| CARA | Control Authority Regulation Architecture — recovery protocol |
| Authority Level | Computed governance authority (0.0-1.0) governing operational actions |
| PRNG | Pseudo-Random Number Generator — Mulberry32 seeded for reproducibility |
| Governance Pipeline | Sequential processing chain: SATA -> HMAA -> MAIVA -> FLAME -> CARA |

## 12.2 References

[1] Oktenli, B. (2026). ADARA Deception Detection Simulation. DOI: 10.5281/zenodo.19043924.

[2] Oktenli, B. (2026). Simulation artifact. https://burakoktenli.com/adara-simulation.html

[3] Oktenli, B. (2026). Research Portfolio. https://burakoktenli.com

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

[5] NIST (2023). AI Risk Management Framework (AI RMF 1.0).

## 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