

Provenance and Security in Space Data Network

Records that name their sources, results anyone can re-run, and screening that checks itself

Anthony “TJ” Koury III and Claude Opus 5.5

Koury: Edgesource, Space Data Network · tj@edgesource.com. Claude: Anthropic.

Technical whitepaper 1.0 | 11 October 2026

Numerical evidence cutoff: 11 October 2026

1 Summary

Two companion papers report the physics: how Space Data Network (SDN) builds and measures its orbital catalog¹, and how it screens that catalog for close approaches². This paper covers what makes those results traceable, checkable and safe to share:

1. **Records.** Every input keeps its provider, edition and the SHA-256 of its bytes, and every solution names the records it rests on (section 2).
2. **Publication and reproduction.** A catalog snapshot is published whole, signed and content-addressed. Every experiment’s numbers trace to a run manifest, and the same modules re-run them in a web browser (section 3).
3. **Screening that checks itself.** Screening work units name every input by hash, and their outputs are byte-identical across runtimes, so replicas check each other. In a local test, a worker that hid conjunctions was caught and its work redone (section 4).
4. **Keys.** A signature is only as good as the key behind it. SDN’s keys carry the security bond of the Adversarial Security paper, which makes a key’s integrity publicly checkable³ (section 5).

2 Records and lineage

A provider’s ephemeris or element set already embeds the provider’s models and inference, so the catalog keeps it as an attributed provider claim. Agreement between products counts as independent confirmation only when their lineage is independent; unknown lineage is recorded as unknown.

A source record keeps the provider, native identifier, edition, SHA-256 of the bytes, product type, epoch, time scale and frame; observation time, solution epoch, issue time and retrieval time stay distinct. A solution record specifies:

Field	Content
State and validity	State or ephemeris, epoch, applicable interval
Evidence and lineage	Source records, their levels and dependencies, editions
Dynamics	Force model, environment inputs, parameters estimated, considered or assumed
Measurement model	Sensor geometry, timing, calibration, biases; unknown when a provider does not state it
Inference	Estimator or selection rule, weights, priors, software version
Uncertainty	Covariance, its source and its calibration label

Hashes establish byte identity and signatures establish attribution; physical validity rests on the measurements of the catalog paper¹. Identical inputs, module binaries and policy reproduce the same decisions.

3 Publication and reproduction

Publication is transactional: acquire and hash the bytes, verify, normalize, validate, sign a manifest, then announce the new catalog head. Consumers read one complete, immutable snapshot; each update names the object, the prior and new solutions, the epoch, the recipe and the provenance. Content addressing and independent replication keep a snapshot retrievable from any node that holds it.

Each experiment's numbers come from a run manifest: the repository and module commits, each module's WebAssembly SHA-256 and the SHA-256 of every input; reports are generated from metrics⁴. The experiments' public site runs the same modules in a web browser: a V1 seed reproduces the committed result exactly, and any Orekit case re-runs from its exact request. Space-Track element sets, SpaceX and Planet files and SET's indices are cited by URL and SHA-256, not copied.

Evidence	Commit
Experiments V1, E1–E8, E13 and E14	orbit-accuracy-experiments 357d992d6c62daac4ccbb2fd230634ac292192b8
HPOP 1.3.0, its Orekit suite, screen and covariance calibration; maneuver detection 0.3.0	space-data-network-modules be9ac0db21cd95d52e184bb45d74d73dcf1f09ac
HPOP in V1	space-data-network-modules 750b6399173a2678b8e8d0885557976d17a9ad5c
VCM adapter	space-data-network-modules 1879d8c5c17c061e157c100723a8423c59fd98e2
Vimpel conversion and refinement	space-data-network-modules 49e159d7003cac3d3e65b03170f11909fa86dd2c

HPOP's WebAssembly binary in V1, SHA-256:

473d527175de6606ccf305c45b37218288634fa9de2dcab0d72564a190bdabd1

4 Screening that checks itself

A screen splits into work units^{2,5}:

- **Grid units:** a block of catalog rows over a time window gives those rows' coarse grid, the screen's index.
- **Screen units:** a block of primary objects over a window gives the events of their pairs. Earlier blocks are skipped and the rest are secondary, so every pair is in exactly one unit.

A unit record names every input by SHA-256: the module, the catalog's element sets, the window's grid blocks and exclusions, the roles and the controls. Its identifier is the SHA-256 of the record. The conjunction module's outputs depend on neither the runtime nor the thread count, so two replicas of a unit must agree byte for byte. Volunteer computing otherwise compares floating-point results only between matched platforms^{6,9}; WebAssembly's determinism⁸, carried through a threaded module, removes that need. Each answer is filed by hash and announced with an SDS publish notification (PNM): the output's content identifier, the unit identifier and an Ed25519 signature.

In a local test, five worker processes acted as independent volunteers, each with its own module instance and key, and declared their threads and memory; a unit went only to a worker with the memory it needed. A keyed 30 % of units ran on two workers, on different runtimes where one was free. A mismatch went to a third worker, the majority won, the losing worker's answers stopped counting, and every unit that relied on them ran again⁷. On the 3-day screen of the 32,514-object catalog, 96 units:

Measure	Result
Replicas compared among honest workers	27, all byte-identical; 23 across runtimes (Node V8 and WasmEdge ahead-of-time)
A worker hiding the closest conjunction of half its screen units	Caught on 5 mismatches; its 9 altered outputs discarded; 13 units run again
A worker leaving 30 % of its units unanswered	Retired after 2 missed deadlines
A worker with 768 MiB	Given only the grid units it could hold
Conjunctions	294,544; the event set's SHA-256 equals the single-process screen's

Replication catches a dishonest worker only through a replicated unit it altered: with replication rate r , m altered units all escape with probability $(1 - r)^m$, 4 % for $r = 0.3$ and $m = 9$. Once caught, all of its work is redone.

An operator's node screens its own fresh data the same way and publishes the data and the results together. Its ephemerides are fitted to SGP4 element sets in memory, its objects' rows replace theirs in the network's grid, one screen unit runs per window, and a PNM signed by the operator's key names the manifest of element sets, unit records and outputs. An independent process in another runtime rebuilds each window's grid from the unit's sources, re-runs each unit and compares bytes. On 11 October 2026, 3 days ahead:

	64 Starlink satellites, SpaceX's files fitted here	The constellation, CelesTrak's fits
From data in memory to the signed PNM	9.1 s, 23 CPU-s	45.9 s, 150 CPU-s
Independent check, WasmEdge ahead-of-time in a separate process	12 of 12 window grids and 12 of 12 outputs byte-identical; signature valid	The same

5 Keys

Records, manifests and notifications are signed, so their worth rests on the keys behind them. Every key an SDN node uses for a purpose is derived from the node's root key unless the operator configures another. The Adversarial Security paper turns such a key into public, continuous evidence of its own integrity³: addresses derived from the key on several blockchains hold value that anyone who steals the key can take, so value left in place shows the key is intact, and the amount bonded sets how much a verifier can rely on it.

Data credits

- Source: USSPACECOM / 18th Space Defense Squadron, via Space-Track.org (<https://www.space-track.org>).
- SpaceX Starlink public ephemerides, <https://api.starlink.com/public-files/ephemerides/> (SpaceX).
- CelesTrak supplemental GP, <https://celestrak.org> (T. S. Kelso).

References

R1

Koury, A. Evidence-Supported ASO Catalog. Space Data Network technical whitepaper 2.0, 11 October 2026. Paper

R2

Koury, A. Fast All-vs-All Conjunction Screening. Space Data Network technical whitepaper 1.9, 11 October 2026. Paper

R3

Koury, A. Persistent Adversarial Security. Space Data Network technical whitepaper. Paper

R4

Edgesource. Orbit accuracy experiments: frozen plans, run manifests, generated reports and data licenses, with a browser site that re-runs them. Commit 357d992d6c62daac4ccbb2fd230634ac292192b8. Repository, site

R5

Edgesource. Continuous, distributed, operator-first screening, 2026-10: work units, replication, the local harness and operator-first publication. Modules commit e1e6577acb0e54eb7f68ba909bc109b7b2a7a77d, **analysis/conjunction-assessment/docs/continuous-distributed-screening-2026-10.md**. Private repository, available on request (tj@edgesource.com).

R6

Anderson, D. P. BOINC: a system for public-resource computing and storage. 5th IEEE/ACM International Workshop on Grid Computing, 4–10, 2004. doi:10.1109/GRID.2004.14

R7

Sarmenta, L. F. G. Sabotage-tolerance mechanisms for volunteer computing systems. Future Generation Computer Systems 18, 561–572, 2002. doi:10.1016/S0167-739X(01)00077-2

R8

Haas, A., et al. Bringing the web up to speed with WebAssembly. PLDI 2017, 185–200. doi:10.1145/3062341.3062363

R9

Taufer, M., et al. Homogeneous redundancy: a technique to ensure integrity of molecular simulation results using public computing. IPDPS 2005. doi:10.1109/IPDPS.2005.247