OrbShield: A Frame-Level Intrusion-Detection System for the CCSDS Telecommand Uplink

Abdulaziz Albarrak **Abdulelah Almutairi** **Saleh Suwaileh**
KAUST Academy, University of Glasgow International Summer School, 2026
Mentor: Prof. Mudassar Aslam

Abstract—OrbShield is a controlled attacker-versus-defender testbed for the CCSDS Telecommand Space Data Link Protocol. It parses and serializes transfer frames, generates seven observable attack scenarios, applies deterministic policy detectors and two supervised baselines, and presents read-only alerts in a dashboard. The threat model separates an uplink injector, which can observe, alter, replay, or inject frames but lacks the managed-security key, from a compromised authorised commanding host, which can create fresh authenticated commands. The second actor is outside the frame monitor's guarantee. Three protocol profiles prevent structural security fields from being mistaken for authentication: clear, a controlled HMAC test security association, and a legacy structural mode. Evaluation uses 20 clean training sessions and 30 disjoint test sessions per condition, with feature state reset between sessions. On unseen clean synthetic sessions, the rules obtain precision 1.000, recall 0.933, and F1 0.966. Under paired loss, duplication, bounded reordering, and evasion, rule F1 falls to 0.605 with FPR 0.735; Random Forest reaches F1 0.848. A controlled one-CLCW-per-frame FARM-like branch lowers FPR to 0.0102 and reaches F1 0.815, but is a simulated upper bound. These values validate project-authored constructions, not operational effectiveness. Local fixtures exercise the byte path; 20 passes over 2,745 frames yield 0.0232 ms median encode-to-classify latency, but do not establish independent field validity. OrbShield therefore contributes a reproducible frame-policy research artifact and an explicit account of its failure boundaries, not an inline blocker or a replacement for CCSDS Space Data Link Security.

Index Terms—CCSDS telecommand, intrusion detection, space cybersecurity, threat modeling, reproducible evaluation.

I. INTRODUCTION

A satellite command path joins operator software, ground networks, radio equipment, a space link, and a spacecraft receiver. Compromise anywhere upstream can convert ordinary ground access into spacecraft input. The Viasat KA-SAT incident illustrated the consequence at ground-network scale: after obtaining access to a trusted management segment, the attacker issued legitimate management commands that overwrote modem flash memory [1]. A controlled ESA exercise on OPS-SAT likewise began with authorised access and progressed to payload compromise [2]. Neither event proves a CCSDS telecommand-frame exploit. They instead motivate the harder question addressed here: what can, and cannot, be inferred by observing the uplink frame layer after a ground-side foothold exists?

CCSDS telecommand standards prioritise predictable interoperation and reliable delivery across missions [3], [4]. Their fields expose useful invariants—declared length, frame error control, destination and virtual channel, sequence state, and optional managed-security material—but those invariants are not equivalent to operator identity or command intent. A frame error-control field detects transmission error; it is not a message authentication code. A fresh command produced by an authorised, key-holding host can therefore be byte-for-byte indistinguishable from a legitimate command at this layer. That limitation is central to the design rather than a footnote.

Prior work is not empty. Zhang et al. published a CCSDS telecommand misbehavior detector in 2010 [5]. ARCA SATLINK provides an end-to-end CCSDS-compliant attack platform [21], and SpaceSecLab provides a broader satellite-security testbed [22]. Recent space-security research has examined protocol fuzzing [6], spacecraft software [7], radio and network paths [8], telemetry anomaly detection [9], and labelled CubeSat data with reproduction scripts [10]. The narrower gap we target is an inspectable teaching and research artifact that combines a TC-frame byte path, documented attacker scenarios, explicit security profiles, deterministic and learned baselines, session-grouped evaluation, and an operational display. We do not claim that allow-lists, counters, rate limits, CRCs, or Random Forests are individually novel.

This work makes four bounded contributions:

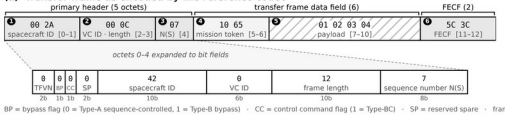
- A CCSDS TC transfer-frame parser/serializer with explicit Type-AD, Type-BD, and Type-BC constraints, managed segment-header decoding, optional FECF validation, and three named security profiles.
- A two-actor threat model, STRIDE analysis, attack tree, and seven reproducible scenarios whose ATT&CK mappings are used only where a current technique is a defensible match.
- Seven interpretable policy detectors and two supervised baselines evaluated on disjoint complete sessions, including paired stress, evasion, confidence intervals, any-alert and correct-attribution recall, ablation, base-rate analysis, and repeated processing-latency measurement.
- A reproducibility package and read-only dashboard that expose raw verdicts, configuration, hashes, and limitations while making no claim of blocking, receiver acceptance, or operational certification.

II. PROTOCOL SCOPE AND CORRECTNESS

A. Transfer-Frame Semantics

OrbShield models the TC Space Data Link transfer frame defined by CCSDS 232.0-B-4 [3]. The primary header contains the transfer-frame version, bypass and control-command flags, spacecraft identifier (SCID), virtual-channel identifier (VCID), declared length, and frame sequence number. Type-AD frames use sequence control and are associated with COP-1 delivery; Type-BD frames bypass sequence control; Type-BC frames convey control commands and do not carry the same user-data structure. The parser rejects the reserved B=0, C=1 combination and rejects security or segment material on Type-BC frames. Type-BD and Type-BC require N(S)=0. A six-octet Type-AD or Type-B frame without FECF is accepted when the managed profile omits FECF, avoiding the earlier global eight-octet minimum. Valid Unlock and Set-V(R) Type-BC directives are governed by a separate explicit recovery-command policy rather than being misclassified as missing security material.

(a) Transfer frame as encoded by the reference codec



(b) Attack surface of each field, and the detector that covers it

	Frame field	Attack class	STRIDE	ATT&CK	Detector (fire order)
❶	destination SCID [0–1]	UC5 Destination/channel policy violation	T		route policy (legacy D3)
❷	VC ID [bits 16–21]	UC5 Destination/channel policy violation	T		route policy (legacy D3)
	frame length [bits 22–31]	UC6 Malformed / fuzzing	T,D		malformed (D1)
❸	frame sequence N(S) [4]	UC1 Telecommand replay	S,T	T0831	replay (D4)
	frame sequence N(S) [4]	UC3 Type-AD continuity anomaly	T		sequence_spoofing (D5)
❹	mission token [5–6]	UC2 Mission-token policy violation	T	T1692.001	injection (D6)
❺	payload [7–10]	*none at the frame layer*	—	—	— (blind spot)
❻	FECF / CRC-16 [11–12]	UC6 Malformed / fuzzing	T,D		malformed (D1)
—	arrival rate [not a field]	UC4 Command flooding (DoS)	D	T0814	flooding (D7)
—	SDLS header [absent here]	UC7 Missing/malformed SDLS field	T		SDLS structure (legacy D2)

ATT&CK techniques: No clean ATT&CK technique; frame-counter anomaly · T0814 Denial of Service
T0831 Manipulation of Control (contextual) · T1692.001 Unauthorized Message: Command Message
Detectors run in the fixed order D1–D7 and the first match wins, so a frame is attributed to the earliest rule it violates.
The payload carries no frame-level signature: a valid, authorised command is byte-identical to a legitimate one, and no frame-layer rule can separate them.
Frame octets emitted by the project codec: 002a800c071865018283845c3c

Fig. 1. Parsed TC frame fields and the observable attack surface. The experimental mission token is not a CCSDS opcode.

When a segment header is configured for a virtual channel, it precedes the managed-security material on the optional frame error-control field (FECF) is checked after decoding. The implementation preserves the full raw byte string and obtains all detector inputs from the decoded representation. Sequence flags and MAP ID are decoded; packet reassembly is not implemented, and continuation/final segments are excluded from mission-token policy because their first octets are not a new token. Thus the headline path is not an in-memory object shortcut: generator → serializer → parser → validator → detector.

B. Experimental Mission Token

The first two octets of the experimental user-data profile are called a mission token. They let the testbed express a mission-specific command allow-list, but CCSDS 232.0 does not define a universal opcode at this location. Whether a real mission can apply this detector depends on its packet or application-layer format. The public API retains a legacy command_code alias for compatibility; the paper does not treat it as a generic CCSDS field.

C. Security Profiles

TABLE I. EXPLICIT SECURITY PROFILES

Profile	Wire treatment	Permitted claim
clear	No security header/trailer	UC7 omitted; structural and policy checks only
verified_test_sa	SPI + counter + truncated HMAC-SHA-256	Controlled authenticity and anti-replay experiments
legacy_structural	Security-shaped bytes, no cryptographic proof	Backward compatibility; never called authentication

The verified_test_sa profile uses a public test key and truncated HMAC-SHA-256 over a defined authenticated region, with a security parameter index and a 64-entry sliding anti-replay window. It is a controlled test security association, not a full implementation or conformance claim for CCSDS 355.0-B-2 [11]. Unknown indices, wrong-length tags, failed tags, and duplicate or stale security counters are rejected. In the clear profile, no authentication-bypass scenario is generated because the supposed evidence does not exist.

III. THREAT MODEL

A. Actors, Assets, and Trust

The protected assets are authorised command content, intended spacecraft/channel selection, freshness, link availability, and the integrity of the monitor's evidence. OrbShield assumes a passive, off-path sensor can see the same TC frames presented to the radio or gateway. The default does not assume access to return-link CLCWs, flight software state, operator identity, or the spacecraft application's execution result.

An optional research branch adds a trusted simulated return path. A bounded FARM-like receiver emits one four-octet CLCW per transmitted frame; OrbShield decodes its VCID, N(R), link/FARM flags, and FARM-B counter before classifying the buffered Type-AD observation. This controlled pairing is not flight telemetry or a complete COP-1 implementation and does not change the uplink-only guarantee.

TABLE II. ADVERSARY CAPABILITY MATRIX

Actor	Capabilities	Boundary
A1a: upstream injector	At or before tap: observe, replay, alter, drop, duplicate, inject; recompute unkeyed FECF	No managed-security key; no trusted monitor configuration
A1b: downstream RF injector	Transmit after the monitor tap	Extra RF frames are invisible unless the sensor also observes that point
A2: authorised-host compromise	Use commanding workflow and keys; send fresh valid frames	Cannot silently change independently protected monitor policy

A1a motivates malformed frames, replays, destination changes, counter manipulation, forged tags, and floods. A1b is explicitly outside the guarantee of a single upstream tap. A2 is the stronger and more realistic consequence of a fully compromised commanding host. If A2 creates a fresh, policy-conforming, correctly authenticated command, all frame-layer detectors should accept it. OrbShield can still reveal subsequent policy violations, but cannot infer malicious intent. Defending A2 requires controls above or beside this layer: multi-party command approval, independent intent validation, least privilege, key separation, host attestation, and spacecraft-side safety checks. SDLS remains valuable against A1a but shares the key-compromise boundary against A2.

B. Attack Tree, STRIDE, and ATT&CK

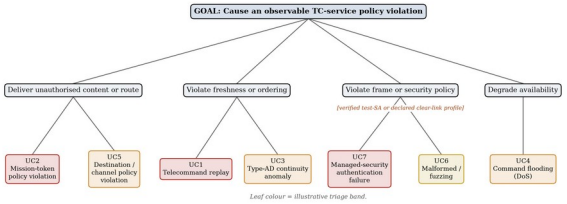


Fig. 2. Attack tree of observable TC-uplink policy violations. Leaf colour denotes an illustrative triage band, not measured probability.

TABLE III. SCENARIOS AND DEFENSIBLE TAXONOMY

UC	Scenario	Observed condition	ATT&CK-ICS	Detector
UC1	Replay	Duplicate frame or security counter	T0831 contextual	Replay / managed security
UC2	Mission-token violation	Token outside configured mission policy	T1692.001	Mission-token policy
UC3	Type-AD continuity	N(S) differs from monitor's local expectation	No clean mapping	Continuity
UC4	Flooding	Trailing-window rate exceeds limit	T0814	Rate
UC5	Destination/ channel	SCID/VCID not configured	No clean mapping	Destination/ channel policy
UC6	Malformed	Length, flags, or FECF invalid	No clean mapping	Parser/validity
UC7	Authentication failure	Known-SA verification fails	No clean mapping	Managed security

The taxonomy is deliberately conservative. MITRE ATT&CK for ICS currently provides a direct command-message sub-technique (T1692.001) and denial-of-service technique (T0814) [12], [13]. Replay is linked contextually to Manipulation of Control (T0831) [14]. Earlier draft mappings to T0820, T0858, and T0859 were removed because those techniques describe exploitation for evasion, changing operating mode, and valid accounts—not malformed frames, link authentication failure, or SCID/VCID fields. An alert demonstrates only an observed condition; it does not prove receiver acceptance or mission effect.

STRIDE labels are used as design prompts, not as quantitative evidence. Likelihood and impact values on a 1–5 scale are multiplied into illustrative triage weights: 1–4 Low, 5–9 Medium, 10–16 High, and 17–25 Critical. These scores help prioritise a demonstration dashboard. A mission would have to replace them with values from its own hazard analysis, command criticality, contact plan, and response procedures.

IV. SYSTEM DESIGN

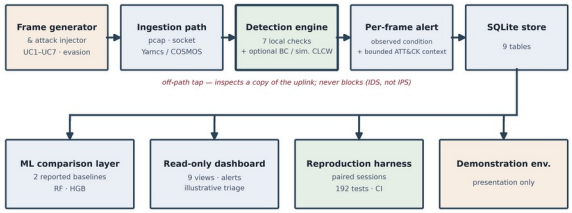


Fig. 3. OrbShield data path. The dashboard is off-path and read-only; it cannot stop a frame already observed.

A. Attacker and Byte Path

A seeded generator creates benign contact sessions and injects labelled variants for the seven scenarios. The serializer computes declared length, optional security material, and FECF. Attack operators then modify the wire-relevant properties appropriate to their scenario; for example, a forged-tag case first creates a valid tag, corrupts it, and recomputes the unkeyed FECF. Replay reuses the original security credentials rather than accidentally creating a fresh authenticated message. Chronological security counters are normalised before injection so benign frames do not become false replays by construction.

B. Defender

The parser and validator first reject impossible or inconsistent structures. The managed-security detector verifies the test association and tracks its anti-replay window. Policy detectors then check whole-frame replay, destination/channel membership, the experimental mission token, exact Type-AD sequence state, and trailing-window rate. Replay identity is the hash of the complete encoded frame, not only N(S) and payload, so repeated fresh Type-BD commands are not collapsed. Type-BD traffic is exempt from COP-1 continuity because its sequence field is not used for sequence control; Type-BC is handled by the optional recovery-command policy and never requires SDLS material. Detection receives the exact parsed evidence and never repairs or re-signs it. The previous heuristic that accepted positive forward gaps as presumed retransmissions was removed: without trusted return-link state, the receiver's acceptance state is unknown.

The deterministic engine has a fixed order and returns the first matching reason. This makes a verdict reproducible but can mask a secondary cause; changing detector order can change attribution even when binary detection is unchanged. The dashboard therefore shows the parser outcome, first policy reason, raw fields, scenario label in demonstration mode, and risk band. It supports triage and later command decisions. It cannot guarantee an alert arrives before spacecraft execution, and no such claim is made.

C. Learned Baselines

Random Forest [18] and histogram gradient boosting, implemented with scikit-learn [19], use the same causal, rule-aligned information available at each frame: structural validity, FECF status, managed-security verification, configured destination/channel and mission-token membership, frame type, declared length, sequence gap, bounded seen-before state, inter-arrival time, and trailing-window rate. The feature extractor resets at every contact-session boundary, and replay history is bounded by the configured 64-frame window. The models do not receive attack labels, future frames, or a global history from the test set.

V. EXPERIMENTAL METHOD

A. Questions and Units

RQ1 asks whether the implementation enforces the declared profile and TC-frame invariants. RQ2 compares rules and learned baselines on unseen complete sessions. RQ3 measures robustness to modeled evasion and benign transport disturbance. RQ4 tests whether the artifact can be rerun and whether local processing latency is small relative to a contact window. The contact session—not the frame—is the independent unit for model split and uncertainty. This avoids the leakage that would occur if adjacent frames from one stateful session appeared in both training and test data [15], [16].

TABLE IV. SESSION-GROUPED EVALUATION DESIGN

Stage	Unit	Detail
Training	20 clean sessions	Seeds 100–119; 16,000 frames total
Clean test	30 disjoint sessions	Seeds 1000–1029; 24,000 frames
Evasion test	30 disjoint sessions	27,610 frames; no adaptation
Paired stress + evasion	30 cloned sessions	27,468 observed frames; attacks unchanged
Uncertainty	Session bootstrap	20,000 resamples; 95% interval
Comparison	Paired sign permutation	20,000 sign draws on session F1 deltas

Each canonical training session contains 500 benign and 300 labelled attack frames (16,000 frames total). Here, canonical means no transport-noise or evasion transform; it does not mean benign-only.

B. Conditions

The clean condition checks whether the implementation recognises its own documented constructions on unseen seeds. The evasion condition adds near-threshold flooding, policy-conforming injected tokens, and other scenario variants that challenge a single exact rule. For the stress condition, each evasion session is cloned after attack generation. The transform drops 294 benign observations, duplicates 152 benign observations, and moves 1,669 observations within blocks of at most three positions. It never drops or duplicates an attack, so the attack identities and denominators are identical between paired evasion and stress sessions. These are controlled stressors, not a calibrated distribution from an operational network. The models are fitted only to the clean training sessions; neither thresholds nor models are tuned on the stress tests. A separately labelled prototype baseline pairs each transmitted frame with one trusted CLCW generated before tap impairment. It is an upper-bound controlled experiment, not an estimate of a downlink with delayed, missing, or unauthenticated reports. A regression sweep varies the renamed diagnostic-only CLCW advance value across 0, 2, 8, and 16 frames. Accepted frames, retransmit reports, later acknowledgements, and lockout retain identical alert decisions at every value; no confusion-matrix cell is controlled by that diagnostic setting.

C. Metrics and Artifact Checks

Binary precision, recall, and F1 are computed over frames, while mean F1, sample standard deviation, and bootstrap intervals are computed over sessions. Per-attack any-alert recall and correct-attribution recall show which scenarios an overall score hides. A feature-group ablation measures dependence on structural/security/policy versus temporal inputs. The reproducibility ledger records configuration, detector order, seeds, model parameters, feature names, raw encoded-frame-and-label digests, predictions, and a curated source-snapshot digest. The verified release collects 201 tests: 200 pass and one external live-stack integration is skipped. Passing project-authored tests is implementation evidence, not independent validation.

VI. RESULTS

TABLE V. PERFORMANCE ON DISJOINT UNSEEN SESSIONS

Condition	Detector	P	R	F1	Attr. macro-F1	Session mean [95% CI]
Clean	Rules	1.000	0.933	0.966	0.987	0.966 [0.966, 0.966]
Clean	RF	0.988	0.982	0.985	0.970	0.985 [0.984, 0.986]
Clean	HGB	0.987	0.984	0.986	0.970	0.986 [0.985, 0.987]
Evasion	Rules	0.938	0.715	0.811	0.866	0.811 [0.811, 0.811]
Evasion	RF	0.985	0.755	0.855	0.803	0.855 [0.853, 0.856]
Evasion	HGB	0.983	0.758	0.856	0.805	0.856 [0.855, 0.857]
Paired stress + evasion	Rules	0.483	0.808	0.605	0.724	0.606 [0.591, 0.621]
Paired stress + evasion	RF	0.968	0.755	0.848	0.782	0.848 [0.847, 0.850]
Paired stress + evasion	HGB	0.966	0.763	0.852	0.782	0.852 [0.851, 0.854]
Paired stress + evasion	Rules w/o continuity	0.982	0.642	0.776	0.769	0.776 [0.776, 0.777]
Paired stress + evasion	Rules + sim. CLCW	0.983	0.696	0.815	0.856	0.815 [0.813, 0.817]

On clean sessions, the rules produce no false positives and F1 0.966, while RF and HGB reach 0.985 and 0.986. This is a construction check, not a surprising scientific result, because generators and rules share the same documented profile. Evasion lowers rule F1 to 0.811. Under paired stress plus evasion, rule precision falls to 0.483 and F1 to 0.605. RF reaches F1 0.848 and HGB 0.852. The RF–rules paired session-F1 difference is +0.242; none of 20,000 sign permutations matched the observed absolute mean, so the finite Monte Carlo result is reported as $p < 1/20{,}001$, not $p=0$.

The simulated CLCW-aware branch changes the stress confusion matrix to TP=8,771, FP=152, TN=14,706, and FN=3,839: precision 0.983, recall 0.696, F1 0.815, attribution macro-F1 0.856, and FPR 0.0102. Relative to strict uplink-only continuity, FPR falls by 98.6%. Compared with disabling continuity, it retains the same 152 false positives while improving recall from 0.642 to 0.696 and F1 from 0.776 to 0.815. UC3 any-alert and attribution recall are both 0.442. These are controlled-emulator results under one paired CLCW per transmitted frame; they do not establish performance with an operational downlink.

TABLE VI. ANY-ALERT AND CORRECT-ATTRIBUTION RECALL UNDER PAIRED STRESS

Scenario	Injected	Rules any	Rules attr.	RF any	RF attr.
UC1 Replay	910	0.997	0.986	0.976	0.975
UC2 Token	1500	0.600	0.582	0.620	0.600
UC3 Sequence	1500	0.872	0.872	0.575	0.560
UC4 Flood	5400	0.700	0.149	0.677	0.649
UC5 Destination	900	1.000	1.000	1.000	1.000
UC6 Malformed	900	1.000	1.000	0.874	0.823
UC7 Authentication	1500	1.000	1.000	1.000	0.600

The per-attack table prevents a misleading interpretation of the aggregate. Rules retain perfect recall for replay, destination/channel violations, malformed frames, and failed authentication in this construction, but any-alert recall is 0.600 for mission-token violations. Flooding is the clearest first-fail masking case: rule any-alert recall is 0.700 but correct-attribution recall is only 0.149 because earlier continuity alerts claim many frames. RF improves flood attribution to 0.649 and sharply reduces false positives, but its UC7 attribution recall is 0.600 despite any-alert recall 1.000. Binary F1 therefore cannot substitute for taxonomy accuracy.

TABLE VII. RANDOM-FOREST ABLATION ON STRESS CONDITION

RF feature group	Recall	F1
All rule-aligned	0.755	0.848
Structural + security + policy	0.971	0.718
Temporal only	0.628	0.705
Policy membership only	1.000	0.629

The ablation confirms that no single feature family explains the best F1. Policy-only inputs maximise recall but generate too many false alarms (F1 0.629). Temporal inputs alone obtain F1 0.705. Removing temporal inputs increases recall to 0.971 but reduces F1 to 0.718. The full rule-aligned set gives the best balance at 0.848.

A. Base-Rate Consequences

The stress condition contains far more attacks than an ordinary command stream. Using its measured false-positive and true-positive rates only as a sensitivity calculation, at one malicious frame per 10,000 and 100,000 frames/day the rules would yield approximately 73,503 alerts/day and positive predictive value (PPV) 0.000110. RF would yield approximately 2,134 alerts/day and PPV 0.00354. These figures are not operational forecasts because the stress distribution is synthetic, but they demonstrate the base-rate problem [17]: even the better model is not ready for unattended operator alerting at low prevalence. Under the same sensitivity calculation, the simulated CLCW rules would yield approximately 1,030 alerts/day and PPV 0.00675. Mission deployment would need calibrated traffic, aggregation, context, and alert-budget targets.

B. Controlled Trace and Latency

Five locally generated Yamcs PCAPs [20] contain 2,745 controlled TC frames across Type-AD, Type-BD, different rates, and diverse configured values. They are useful byte-path fixtures but lack independent capture provenance and must not be described as operational satellite traffic. On the revision machine, full encode → decode → classify latency was measured after one discarded warm-up across 20 complete passes (54,900 frame observations): 0.0232 ms median, 0.0256 ms at p95, 0.0361 ms at p99, and 0.5698 ms maximum. Hardware, capture, network, dashboard, operator-decision, and command-round-trip latency are excluded. The measurement shows local computation is not the bottleneck; it does not prove an alert precedes spacecraft execution.

VII. WHAT THE RESULTS DO AND DO NOT ESTABLISH

The clean result establishes that the parser, scenario generators, and detectors agree on the declared experimental profile across unseen seeds. The stress result establishes that this agreement is fragile: exact rules are interpretable, but benign disturbance can make their alerts unusable. The learned baselines absorb some of that variation, yet their evidence remains internal to one generator family. Neither result estimates performance on another mission, another implementation, an RF channel, or a hostile key-holding operator.

The simple rule engine still has value. It provides executable protocol assertions, transparent causes, a regression oracle for parser changes, and a low-complexity baseline against which learned models must earn their additional uncertainty. It should not be defended by claiming superior robustness; the measured stress result shows the opposite. A credible operational design would likely combine cryptographic verification, exact conformance checks, state reconciled with trusted return-link information, and a calibrated anomaly layer.

The dashboard's operational value is retrospective and forward-looking. After an alert, an operator may stop subsequent commands, isolate a ground host, rotate a security association, compare return-link state, or request spacecraft safe-mode procedures. The monitor is not inline, so the triggering frame may already have been received or executed. Presentation language and labels are constrained to

'observed', 'detected', and 'requires investigation' rather than 'blocked' or 'prevented'.

VIII. LIMITATIONS AND VALIDITY THREATS

Protocol coverage. The artifact is a research subset of CCSDS 232.0. Its CLCW branch uses a bounded FARM-like emulator and one paired report per frame; it does not implement the complete COP-1 state machines, operational telemetry transport, coding/synchronisation layers, RF effects, or application command semantics.

Type-BC coverage. Valid Unlock and Set V(R) directives are parsed and can be checked against an explicit local recovery policy, but the V4 benchmark contains no malicious Type-BC instances and therefore does not quantify their detection.

Security coverage. The HMAC test association is intentionally simpler than CCSDS 355.0 and is not a CryptoLib or standards-conformance substitute. Key management and algorithm agility are out of scope.

Intent and identity. SCID/VCID identify a destination/channel, not an operator. A key-holding authorised host can create an undetectable malicious command. The mission-token policy is an experimental application convention.

Data realism. Synthetic attacks and detectors were co-designed. Whole-session separation removes frame leakage but does not create organisational or mission diversity. Controlled Yamcs traces are not independent operational captures.

Noise model. Loss, duplication, and reordering probabilities are stress parameters, not estimates from a flight network. PPV calculations are sensitivity analyses rather than forecasts.

Attribution. First-fail ordering can hide secondary causes. Binary detection and scenario attribution must be reported separately.

Risk and outcome. Triage weights are illustrative. No experiment observes receiver acceptance, application execution, physical consequence, or operator response quality.

Deployment. The dashboard is off-path. Measured Python processing time excludes capture and human response; there is no availability, failover, tamper-resistance, or certification study.

IX. REPRODUCIBILITY AND RELEASE

The artifact package contains parser, serializer, profile configuration, attack generators, deterministic detectors, learned baselines, dashboard/server, tests, controlled PCAP fixtures, threat-model documents, and a single evaluation entry point (secondsource/reproduce_v4.py). A container entry point invokes the same grouped evaluation. The ledger binds configuration, detector order, sessions, frame bytes and labels, model parameters, and prediction results with SHA-256. Release orbshield-v4-20260812 contains 271 curated source/evidence files; its source-snapshot SHA-256 is 32D868EFD35164B359A036580DCB72FA94BDE89E593D8BF60421B892E890B019, and its result-ledger SHA-256 is CC9BA2A1BE0E2C045602E2FA9C6AB56699F0EA3D4EE4330B1E134153370432D1. The archive is locally reproducible and was rerun from that snapshot, but no public repository URL or DOI is claimed.

X. CONCLUSION

OrbShield recreates a bounded set of CCSDS telecommand-uplink security challenges as a reproducible attacker–defender exercise. It now distinguishes structural fields from verified authentication, respects the service-type boundary, limits ATT&CK claims, evaluates complete unseen sessions, and reports failure under noise rather than hiding it. The deterministic rules are useful as transparent conformance and policy assertions; the learned baselines are more robust in the modeled stress condition but remain internally validated prototypes. A simulated CLCW return path demonstrates that receiver-state correlation can remove most continuity false alerts while recovering some UC3 recall, but operational CLCW transport remains unvalidated. Most importantly, the threat model states the irreducible boundary: a fresh valid command from a compromised authorised host cannot be identified from TC-frame bytes alone. Future work should add independent mission traces, a conformant SDLS implementation, trusted CLCW/COP-1 state, cross-layer command intent, inline response experiments, and a preregistered multi-mission evaluation.

ACKNOWLEDGMENT

The authors thank Prof. Mudassar Aslam and the KAUST Academy / University of Glasgow International Summer School mentors for project supervision and review.

REFERENCES

[1] Viasat, “KA-SAT network cyber attack overview,” Mar. 30, 2022. [Online]. Available: https://www.viasat.com/perspectives/corporate/2022/ka-sat-network-cyber-attack-overview/

[2] European Space Agency, “ESA oversees in-orbit cybersecurity demonstration,” May 4, 2023. [Online]. Available: https://www.esa.int/Enabling_Support/Operations/ESA_oversees_in-orbit_cybersecurity_demonstration

[3] Consultative Committee for Space Data Systems, TC Space Data Link Protocol, CCSDS 232.0-B-4, Issue 4, Oct. 2021, Corrigendum 1, Oct. 2023.

[4] Consultative Committee for Space Data Systems, Communications Operation Procedure-1, CCSDS 232.1-B-2, Issue 2, Sep. 2010, Corrigendum 1, Apr. 2019.

[5] L. Zhang, C. An, Q. Zhang, and C. Tang, “Misbehavior detection algorithm in CCSDS space telecommand system,” IEEE Commun. Lett., vol. 14, no. 8, pp. 746–748, Aug. 2010, doi: 10.1109/LCOMM.2010.08.091893.

[6] S. Havermans, L. Baumgärtner, J. Roberts, M. Wallum, and J. Caballero, “Fuzzing space communication protocols,” in Proc. Workshop on Security of Space and Satellite Systems, 2025, doi: 10.14722/spacesec.2025.23012.

[7] J. Willbold, M. Schloegel, M. Vögele, M. Gerhardt, T. Holz, and A. Abbasi, “Space Odyssey: An experimental software security analysis of satellites,” in Proc. IEEE Symp. Security and Privacy, 2023, doi: 10.1109/SP46215.2023.10179393.

[8] J. Pavur, D. Moser, M. Strohmeier, V. Lenders, and I. Martinovic, “A tale of sea and sky: On the security of maritime VSAT communications,” in Proc. IEEE Symp. Security and Privacy, 2020, pp. 1384–1400, doi: 10.1109/SP40000.2020.00056.

[9] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, “Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding,” in Proc. ACM SIGKDD, 2018, pp. 387–395, doi: 10.1145/3219819.3219845.

[10] Y. Fayyaz, S. N. Afrasiabi, L. Yang, and K. El-Khatib, “CubeSat cybersecurity dataset for intrusion detection (CuCD-ID),” Data in Brief, vol. 65, Art. no. 112598, 2026, doi: 10.1016/j.dib.2026.112598.

[11] Consultative Committee for Space Data Systems, Space Data Link Security Protocol, CCSDS 355.0-B-2, Issue 2, Aug. 2022. [Online]. Available: https://public.ccsds.org/Pubs/355x0b2.pdf

[12] MITRE, “Unauthorized Message: Command Message, T1692.001,” ATT&CK for ICS, Apr. 2026. [Online]. Available: https://attack.mitre.org/techniques/T1692/001/

[13] MITRE, “Denial of Service, T0814,” ATT&CK for ICS. [Online]. Available: https://attack.mitre.org/techniques/T0814/

[14] MITRE, “Manipulation of Control, T0831,” ATT&CK for ICS. [Online]. Available: https://attack.mitre.org/techniques/T0831/

[15] D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, and K. Rieck, “Dos and don’ts of machine learning in computer security,” in Proc. 31st USENIX Security Symp., 2022, pp. 3971–3988.

[16] R. Sommer and V. Paxson, “Outside the closed world: On using machine learning for network intrusion detection,” in Proc. IEEE Symp. Security and Privacy, 2010, pp. 305–316, doi: 10.1109/SP.2010.25.

[17] S. Axelsson, “The base-rate fallacy and the difficulty of intrusion detection,” ACM Trans. Inf. Syst. Security, vol. 3, no. 3, pp. 186–205, Aug. 2000, doi: 10.1145/357830.357849.

[18] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[19] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[20] Yamcs, “The open-source mission control framework.” [Online]. Available: https://yamcs.org/

[21] L. Masson, M. Bonjour, L. Thoeny, and S. Willy, “Developing a CCSDS compliant platform to reliably secure current and future space data links,” CYSEC, 2023. [Online]. Available: https://indico.esa.int/event/528/attachments/5988/10191/Developing_a_CCSDS_compliant_platform_to_reliably_secure_current_and_future_space_data_links.pdf

[22] D. Fischer, M. Spada, and D. Koisser, “SpaceSecLab: A representative, modular environment for prototyping and testing space-link security protocols end to end,” in Space Operations: Contributions from the Global Community, 2017, pp. 375–394, doi: 10.1007/978-3-319-51941-8_18.