International Journal of Transport Development and Integration
https://www.acadlore.com/journals/IJTDI

Security Analysis and Hardened Redesign of TrustCNAV for Reliable GNSS Civil Navigation Message Authentication in Transport Systems

Haewon Byeon*

Department of Future Technology, Korea University of Technology and Education (KOREATECH), 31253 Cheonan, Republic of Korea

* Correspondence: Haewon Byeon (bhwpuma@naver.com)

Received: 03-05-2026 **Revised:** 04-09-2026 **Accepted:** 04-16-2026

Citation: H. Byeon, "Security analysis and hardened redesign of TrustCNAV for reliable GNSS civil navigation message authentication in transport systems," *Int. J. Transp. Dev. Integr.*, vol. 10, no. 2, pp. 373–384, 2026. https://doi.org/10.56578/ijtdi100204.

Abstract: Global Navigation Satellite Systems (GNSS) civil navigation messages (CNAVs) remain vulnerable to spoofing and meaconing due to their open broadcast nature. TrustCNAV, originally proposed as a certificateless aggregate authentication scheme, aims to provide efficient verification with low receiver overhead. However, its practical robustness under realistic deployment conditions remains insufficiently examined. This study presents a systematic security reassessment and a hardened redesign of TrustCNAV with particular attention to transport-relevant constraints. The analysis identifies critical vulnerabilities, including signing-key exposure under nonce reuse and forgery risks arising from unauthenticated public-parameter updates. To address these issues, an improved protocol variant is developed, incorporating deterministic nonce generation, authenticated parameter distribution, and epoch-consistent batch verification. In addition to protocol redesign, a bounded symbolic trace-exploration approach is introduced to evaluate the security properties of both the original and improved schemes. A communication overhead model at the bit level is also established to reflect CNAV bandwidth constraints. The results indicate that the improved design effectively mitigates the identified vulnerabilities while maintaining a pairing-free structure and acceptable computational cost. The findings highlight the importance of integrating protocol security with system-level considerations, particularly in transport environments where authentication delay and failure may directly affect operational safety.

Keywords: Global Navigation Satellite Systems authentication; Civil navigation message; Certificateless signature; Aggregate authentication; Transport system security; Spoofing resilience

1 Introduction

Global Navigation Satellite Systems (GNSS) underpin safety-critical transport. Air traffic management, maritime navigation, rail operations, and logistics platforms depend on Positioning, Navigation, and Timing (PNT) derived from broadcast civil navigation message (CNAV) signals. The operational risk is not limited to jamming. A capable spoofer can inject forged navigation data while sustaining signal tracking, which turns a cyber intrusion into a physical hazard. TrustCNAV [1] targets this threat by proposing certificateless aggregate authentication for CNAV, aiming for immediate verification and low receiver workload without pairing operations.

The transport relevance of CNAV authentication is not uniform across application domains. In railway train-control architectures, forged or replayed navigation data can distort train-location reporting and degrade movement-authority logic. In aviation and uncrewed-aircraft settings, the more immediate concern is whether the receiver degrades safely when authenticated and non-authenticated data coexist or when message validation arrives too late for time-critical decision loops. In autonomous road systems, strict authentication may improve trustworthiness yet also reduce availability if only a subset of signals can be authenticated in real time. These scenario-dependent trade-offs mean that protocol latency, update trust, and failure handling matter as much as signature algebra [2–4]. For concreteness, the engineering interpretation in this paper centers on GNSS-based railway positioning, where authenticated CNAV screening can directly affect train-location reporting, movement-authority logic, and the decision to accept, reject, or degrade a suspect navigation update [4].

Work on GNSS threats and defenses shows why protocol guarantees should be tested against realistic field assumptions. Psiaki and Humphreys separated meaconing from structured synthesis and highlighted that defenses must accommodate receiver constraints and non-ideal propagation [5]. Humphreys argued that cryptographic checks matter only when the receiver also enforces time-consistency and a detection logic, because the attacker can exploit timing semantics without breaking message integrity [2]. Surveys confirm that spoofing equipment keeps getting cheaper, while GNSS receivers appear in more connected, software-managed transport systems that expand the attack surface [6].

Signal-level defenses such as distortion monitoring and correlation tests can raise the attacker's cost, but their effectiveness varies with antenna configuration, multipath, and front-end quality [7]. Data-level defenses provide an explicit accept/reject gate at the navigation-message layer. GNSS authentication proposals built on unpredictable signal structures, delayed disclosure, and message authentication have clarified the security-latency trade space and have motivated deployment programs such as Open Service Navigation Message Authentication (OSNMA)-like mechanisms [3, 8–14]. At the same time, replay and distance-decreasing strategies remain practical in transport contexts because they target time-of-transmission and receiver policy rather than hash collisions or discrete logs [3, 10].

TrustCNAV's key claim is that a certificateless, pairing-free signature structure can authenticate CNAV efficiently, and that aggregation across satellites reduces total verification time at the receiver [1]. This fits operational needs: receivers often validate multiple satellite messages per epoch, and even small per-message savings can accumulate into meaningful delay reductions. Still, transport deployments rarely satisfy the assumptions used in standard reductions. Ground segments can leak secrets. Parameter updates can travel over maintenance channels with uneven authentication. Implementation faults, especially nonce failures, can break otherwise sound algebra.

This paper presents an independent security analysis and hardened redesign of TrustCNAV. We examine whether a pairing-free certificateless CNAV authentication scheme remains trustworthy when transport-relevant deployment assumptions are relaxed, especially under nonce faults, unauthenticated public-parameter updates, and mixed-epoch buffering. To improve the practical relevance of the study, we also relate the protocol discussion to transport environments in which authentication delay or failure can affect rail positioning, aviation timing logic, or automated road navigation policies.

Our contributions are therefore: (i) an explicit reconstruction of TrustCNAV's message flow and signature equations; (ii) vulnerability proofs with concrete attack traces and clarified deployment assumptions; (iii) a redesign that strengthens binding, nonce robustness, authenticated parameter distribution, and epoch invariants while keeping pairing-free verification; and (iv) a evaluation including communication overhead, transport-scenario interpretation, and bounded symbolic trace exploration for the specific attack classes studied here; the overall workflow appears in Figure 1.

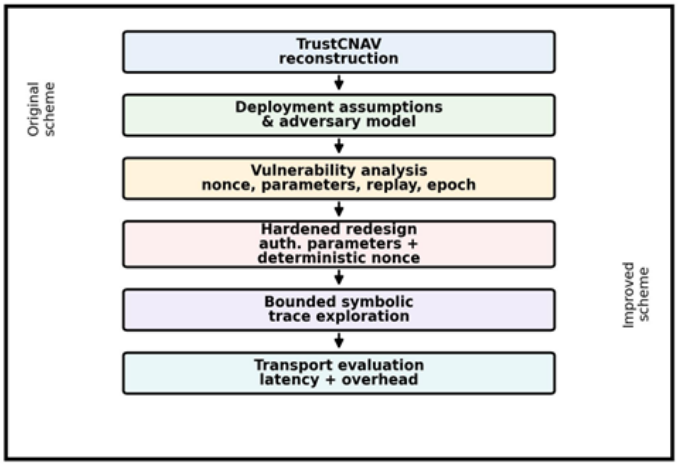

Figure 1. Protocol & analysis workflow

2 Related Works

GNSS spoofing analysis has long emphasized that authentication and detection interact: message integrity does not automatically yield a safe navigation solution unless the receiver constrains timing and consistency [2]. Psiaki and Humphreys cataloged the attack space, including replay/meaconing and structured synthesis, and argued that defenses must respect receiver cost and antenna constraints [5]. Surveys further document how spoofing threats

evolve with software-defined radios and the growing integration of GNSS into Internet-of-Things (IoT)-like transport control stacks [6].

Navigation message authentication research has evolved along two broad lines. One line attaches cryptographic tags or signatures to navigation data, including one-way hash-chain variants and signature-based authentication in BeiDou-style settings [15, 16]. Another line leverages system structure through unpredictable data and delayed disclosure to provide origin authentication with deployment-friendly overhead. Studies of OSNMA testing, CHIMERA/OSNMA integration, and unpredictability mapping illustrate the practical limits of delayed disclosure, especially around time-to-first-authenticated-fix and receiver synchronization [9, 11, 12]. Parameter-selection work for delayed-disclosure protocols highlights the security-latency balance when receivers must wait for key release [13], and semi-assisted designs aim to reduce this waiting time under realistic link conditions [14]. Replay detection remains contentious: partial-correlation detection and worst-case attacker models show that a determined adversary can trade distortion against delay and can exploit mixed authenticated and non-authenticated environments during transitional deployments [10, 17, 18].

Recent developments indicate that GNSS navigation-message authentication is moving from proof-of-concept to operational and field-assessed deployment. The Galileo OSNMA Initial Service is now documented through an operational service-definition framework [19], and recent field work has examined OSNMA behavior under interference-affected radio frequency (RF) road environments [20]. At the same time, transport-specific studies show that spoofing materially affects GNSS-based railway train-positioning architectures and that resilience depends on how message authentication interacts with broader system design [4]. These developments sharpen the practical question addressed in this paper: whether a low-latency CNAV authentication protocol remains trustworthy when deployment assumptions about keys, updates, and batching are stressed.

Certificateless authentication and aggregation appear widely in transport networks outside GNSS. Vehicular Ad Hoc Networks (VANETs) use certificateless mechanisms to reduce certificate management overhead while retaining conditional privacy and traceability goals [21]. Pairing-free certificateless aggregate signatures and key-insulated certificateless authentication in IoT environments highlight recurring pitfalls: poorly authenticated public-key distribution can enable replacement attacks, and implementation faults can negate reduction proofs [22, 23]. Blockchain-based accountability layers have been proposed for GNSS message distribution, though they introduce operational complexity and depend on connectivity assumptions that do not always hold for avionics [24].

TrustCNAV sits at the intersection of immediate verification, certificateless keying, and aggregation [1]. Its novelty lies in binding CNAV structure to a pairing-free aggregate signature equation and claiming low receiver consumption [1]. Our manuscript focuses on the parts that determine real-world trust: authenticity of public parameters, robustness of nonce generation, and epoch-consistent aggregation policies.

3 Methodology

We analyze the TrustCNAV protocol of Wu et al. [1] as an executable message-binding mechanism, then validate the redesign with a bounded symbolic trace-exploration layer. First, we extract the broadcast flow and rewrite it as explicit message instances M_i where each CNAV packet carries (ID_i, m_i, T_i, F_i, SIG_i) and the receiver recomputes hashes and elliptic-curve relations; this makes parsing and canonical encoding part of the security claim. Second, we model the key schedule by separating KGC-issued components from Ground Control Station (GCS) secrets, representing them as compositional terms that can be substituted under compromise (Mas-Key, Pri-O_ID$_i$, Pub-T_ID$_i$) while keeping public artifacts (Mas-Pub, Pub-O_ID$_i$, Pub-T_ID$_i$, Pub-Key_ID$_i$) as attacker-observable. Third, we define adversary capabilities aligned with transport operations: an external spoofer who records and replays CNAV; a network attacker who can inject alternative public parameters; an insider who can extract secrets from the ground segment; and a fault-capable attacker who can induce nonce repetition during signing. Fourth, we set protocol invariants that must hold for safety, including (a) per-message authenticity before aggregation, (b) epoch consistency of all messages admitted into an aggregate check, and (c) non-recoverability of long-term signing keys from signature transcripts. Fifth, we search for violating traces through algebraic substitution and transcript construction.

3.1 Bounded Symbolic Trace-Exploration Procedure

In addition to these manual steps, we add a bounded symbolic trace-exploration layer rather than claiming a full mechanized proof. We describe the protocol using the following event predicates and safety query form:

$$\begin{aligned} &\mathrm{Send}(\mathrm{ID}_i, \mathrm{epoch}, m_i), \quad \mathrm{Accept}(\mathrm{ID}_i, \mathrm{epoch}, m_i) \\ &\mathrm{Accept}(\mathrm{ID}_i, \mathrm{epoch}, m_i) \Rightarrow \exists\, \mathrm{Send}(\mathrm{ID}_i, \mathrm{epoch}, m_i) \\ &\mathrm{State} = (\mathrm{Buffer}, \mathrm{ParamStore}, \mathrm{HonestSends}) \\ &\mathrm{State} = \mathrm{State} \cup (\mathrm{ReplayCache}, \mathrm{FaultFlags}, \mathrm{AcceptedSet}) \end{aligned} \tag{1}$$

The transition system explores honest broadcast, replay, parameter replacement, nonce-fault injection, epoch selection, batch verification, and optional fallback individual verification. To keep the search tractable, hash-equivalent states are merged, duplicate transcript prefixes are pruned, and exploration stops at user-defined limits on depth, batch size, and attacker actions. Under these bounds, the procedure finds the expected accepting traces against the original TrustCNAV variant and finds no accepting traces for the improved variant under the same modeled capabilities. We present these results as bounded symbolic evidence for the specific trace classes studied here, not as a general proof comparable to ProVerif or Tamarin [25, 26].

To keep the description consistent with the transport context, we also use standard modeling idioms as templates for key and trust binding:

$$\begin{aligned} K_{\mathrm{PUF}} &= \mathrm{h}(\mathrm{VID}_i \parallel R_i) \\ SK^{j}_{\mathrm{sess}} &= \mathrm{KDF}\Big(SK_{\mathrm{reg}}, N^{j}_{\mathrm{RS}}, N^{j}_{\mathrm{U}}, \mathrm{context}_j\Big) \\ A_n(t+1) &= A_n(t) + u_n(e_t) \\ p_{\mathrm{safe}}(n, t+1) &= f(p_{\mathrm{safe}}(n,t), e_t) \end{aligned} \tag{2}$$

These examples illustrate a device-derived seed without exposing a raw identifier, a session key bound to nonces and a context label, and trust updates based only on cryptographically verified events. They do not claim that TrustCNAV itself uses PUFs or game-theoretic utilities.

4 Analysis of the Protocol

4.1 Overview of the Proposed Protocol

TrustCNAV is a broadcast authentication protocol for CNAV. Entities include a Key Generation Center/Key Management Center (KGC/KMC), a GCS, satellites S_i, and receivers [1]. The scheme operates over an elliptic-curve group with base point P and master secret Mas-Key held by the KGC. The system parameters are summarized as:

$$\begin{aligned} \text{Sys-Par} &= (P, p, q, E, G, \mathrm{H}_1, \mathrm{H}_2, \mathrm{H}_3, \text{Mas-Pub}) \\ \text{Mas-Pub} &= \text{Mas-Key} \cdot P, \quad \mathrm{H}_1, \mathrm{H}_2, \mathrm{H}_3 : \{0,1\}^* \to \mathbb{Z}_p \end{aligned} \tag{3}$$

Provisioning proceeds per satellite identity ID_i. The KGC selects c_i in $\mathbb{Z}_p$ and computes the partial satellite values as follows:

$$\begin{aligned} \text{Pub-O\_ID}_i &= c_i \cdot P \\ h_{1,i} &= \mathrm{H}_1(\mathrm{ID}_i, \text{Pub-O\_ID}_i, \text{Mas-Pub}) \\ \text{Pri-O\_ID}_i &= (c_i + \text{Mas-Key} \cdot h_{1,i}) \bmod p \end{aligned} \tag{4}$$

It transmits (ID_i, Pub-O_ID$_i$, Pri-O_ID$_i$) to the GCS. The GCS chooses Pub-T_ID$_i$ in $\mathbb{Z}_p$ and computes:

$$\begin{aligned} E_i &= \text{Pri-T\_ID}_i \cdot P \\ h_{2,i} &= \mathrm{H}_2(\mathrm{ID}_i, E_i) \\ \text{Pub-T\_ID}_i &= \text{Pub-O\_ID}_i + h_{2,i} \cdot E_i \\ \text{Pub-Key\_ID}_i &= (\text{Pub-T\_ID}_i, \text{Pub-O\_ID}_i) \\ \text{Pri-Key\_ID}_i &= (\text{Pri-O\_ID}_i, \text{Pri-T\_ID}_i) \end{aligned} \tag{5}$$

For each CNAV payload m_i at time T_i, the signer samples f_i in $\mathbb{Z}_p$ and computes the following values:

$$\begin{aligned} F_i &= f_i \cdot P \\ h_{3,i} &= \mathrm{H}_3(\mathrm{ID}_i, m_i, \text{Pub-Key\_ID}_i, F_i, T_i) \\ \mathrm{SIG}_i &= (f_i + h_{3,i} \cdot (\text{Pri-O\_ID}_i + h_{2,i} \cdot \text{Pri-T\_ID}_i)) \bmod p \\ \sigma_i &= (F_i, \mathrm{SIG}_i), \quad \mathrm{M}_i = (\mathrm{ID}_i, m_i, T_i, F_i, \mathrm{SIG}_i) \end{aligned} \tag{6}$$

The receiver checks freshness via a local window on T_i, recomputes $h_{1,i}$ and $h_{3,i}$, and verifies the individual signature equation:

$$\mathrm{SIG}_i \cdot P = F_i + h_{3,i} \cdot (\text{Pub-T\_ID}_i + h_{1,i} \cdot \text{Mas-Pub}) \tag{7}$$

For aggregation across satellites in the same positioning epoch, it computes:

$$\begin{aligned} F_{\text{sum}} &= \sum_i F_i \\ \text{SIG}_{\text{sum}} &= \left(\sum_i \text{SIG}_i \right) \bmod p \end{aligned} \tag{8}$$

The aggregate verification equation is:

$$\begin{aligned} \text{SIG}_{\text{sum}} \cdot P &= F_{\text{sum}} + \sum_i h_{3,i} \cdot Q_i \\ Q_i &= \text{Pub-T\_ID}_i + h_{1,i} \cdot \text{Mas-Pub} \end{aligned} \tag{9}$$

4.2 Identified Vulnerabilities

TrustCNAV's algebraic structure supports efficient verification, but several implementation- and deployment-facing issues can break its intended guarantees.

Nonce reuse collapses the effective signing key. The signature scalar follows:

$$\begin{aligned} \text{SIG}_i &= (f_i + h_{3,i} \cdot \text{SK}_i) \bmod p \\ \text{SK}_i &= (\text{Pri-O\_ID}_i + h_{2,i} \cdot \text{Pri-T\_ID}_i) \bmod p \end{aligned} \tag{10}$$

Because the nonce is exposed through the transmitted elliptic-curve point, reuse is observable when the same satellite identity emits two broadcasts with the same F_i but different (m_i, T_i):

$$F_i = f_i \cdot P, \quad h_{3,i} \neq h'_{3,i} \tag{11}$$

Under that condition, an eavesdropper who records the two signature transcripts obtains:

$$\begin{aligned} \text{SIG}_i - \text{SIG}'_i &= \big((h_{3,i} - h'_{3,i}) \cdot \text{SK}_i \big) \bmod p \\ \text{SK}_i &= \big((\text{SIG}_i - \text{SIG}'_i) \cdot \text{Inv}(h_{3,i} - h'_{3,i}) \big) \bmod p \end{aligned} \tag{12}$$

Once SK_i is recovered, the attacker forges signatures for arbitrary CNAV payloads m^* by choosing f^* with F^* = $f^* \cdot P$, recomputing ${h_3}^*$, and outputting:

$$\begin{aligned} h_3^* &= \text{H}_3(\text{ID}_i, m^*, \text{Pub-Key\_ID}_i, F^*, T^*) \\ \text{SIG}^* &= \big(f^* + h_{3,i}^* \cdot \text{SK}_i \big) \bmod p \end{aligned} \tag{13}$$

Public-parameter replacement enables key-substitution traces when parameter acquisition or update is not strongly authenticated. Receivers typically learn Pub-O_ID_i and Pub-T_ID_i through operational channels, cached updates, or maintenance interfaces. If an attacker can replace these values for some ID_i, the verification equation becomes a lever. For a substituted key pair, the receiver computes:

$$\begin{aligned} h'_{1,i} &= \text{H}_1\big(\text{ID}_i, \text{Pub-O\_ID}'_i, \text{Mas-Pub}\big) \\ \text{SIG}_i \cdot P &\stackrel{?}{=} F_i + h_{3,i} \cdot \big(\text{Pub-T\_ID}'_i + h'_{1,i} \cdot \text{Mas-Pub}\big) \end{aligned} \tag{14}$$

An attacker who controls the substituted parameters can pick any scalar x, set the substituted public value, and craft an accepting transcript as follows:

$$\begin{aligned} \text{Pub-T\_ID}'_i &= x \cdot P - h'_{1,i} \cdot \text{Mas-Pub} \\ F &= f \cdot P, \quad \text{SIG} = (f + h_{3,i} \cdot x) \bmod p \end{aligned} \tag{15}$$

This attack therefore targets deployments in which receivers obtain or refresh public parameters without a pinned trust anchor. If those parameters are already distributed in a signed package authenticated by the receiver, the attack does not apply; the problem is that the original TrustCNAV description does not fully specify such an authenticated operational path.

(i) KGC compromise expands into universal forgery under realistic leakage. The certificateless model splits trust between the KGC and the GCS. Yet the partial private value embeds Mas-Key linearly, and the public relation makes the dependency visible:

$$\begin{aligned} \text{Pri-O\_ID}_i &= (c_i + \text{Mas-Key} \cdot h_{1,i}) \bmod p \\ \text{Pri-O\_ID}_i \cdot P &= \text{Pub-O\_ID}_i + h_{1,i} \cdot \text{Mas-Pub} \end{aligned} \tag{16}$$

(ii) If Mas-Key leaks, a Type-II attacker can recompute Pri-O_ID$_i$ for any identity after observing Pub-O_ID$_i$. If, in addition, Pub-T_ID$_i$ leaks from the ground segment, the attacker reconstructs:

$$\mathrm{SK}_i = (\text{Pri-O\_ID}_i + h_{2,i} \cdot \text{Pri-T\_ID}_i) \bmod p \tag{17}$$

(iii) The attacker can then forge as in the nonce-reuse attack.

Freshness checks can be bypassed by replay within acceptance windows and by distance-decreasing traces. Each broadcast includes a timestamp T_i, and the receiver checks:

$$|T_{\text{local}} - T_i| \leq \Delta T \tag{18}$$

A meaconer who records the broadcast message and re-broadcasts it within the accepted window preserves signature validity:

$$\begin{aligned} \mathrm{M}_i &= (\mathrm{ID}_i, m_i, T_i, F_i, \mathrm{SIG}_i) \\ \mathrm{SIG}_i \cdot P &= F_i + h_{3,i} \cdot (\text{Pub-T\_ID}_i + h_{1,i} \cdot \text{Mas-Pub}) \end{aligned} \tag{19}$$

Even when ΔT is tight, the attacker can shift perceived time-of-transmission in ways that alter pseudorange. The present redesign, therefore does not claim to defeat distance-decreasing attacks; it only narrows replay and batching abuse at the message-verification layer.

(iv) Aggregate authentication can accept mixed-epoch sets unless the receiver enforces per-message invariants. The aggregate check uses:

$$\begin{aligned} \mathrm{SIG}_{\text{sum}} = \left(\sum_i \mathrm{SIG}_i \right) \bmod p, \quad F_{\text{sum}} = \sum_i F_i \\ \mathrm{SIG}_{\text{sum}} \cdot P = F_{\text{sum}} + \sum_i h_{3,i} \cdot Q_i \\ Q_i = \text{Pub-T\_ID}_i + h_{1,i} \cdot \text{Mas-Pub} \end{aligned} \tag{20}$$

(v) This equation remains valid even if the set includes messages from different epochs, provided the attacker supplies the corresponding tuples (F_i, SIG$_i$, T_i, m_i). Figure 2 sketches the original three-layer framework in which such cross-epoch mixing can happen at the receiver interface when broadcast data are buffered.

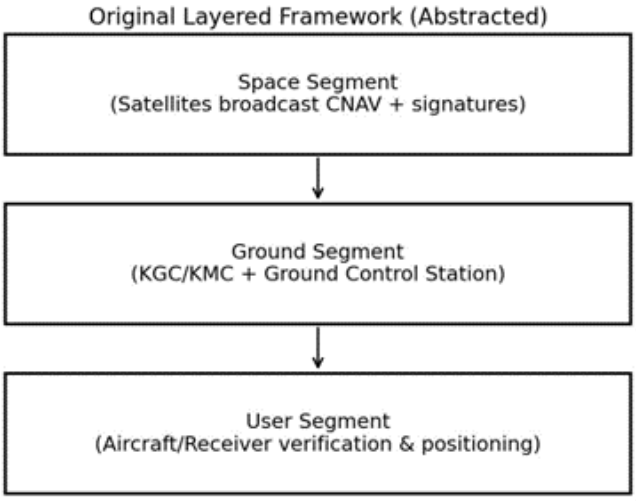

Figure 2. Abstracted layered framework for Trust civil navigation messages (TrustCNAV) deployments

(vi) Missing point validation and subgroup checks create denial-of-service and edge-case acceptance risks. The verification equations assume that points F_i, Pri-O_ID$_i$, and Pub-T_ID$_i$ lie on the correct curve and have the correct order. If the receiver accepts malformed points, an attacker can inject $F_i = \infty$ (point at infinity) or a small-subgroup element. Figures 3 and 4 summarize the main attack surfaces across the satellite-ground-receiver path, including parameter substitution and replay injection.

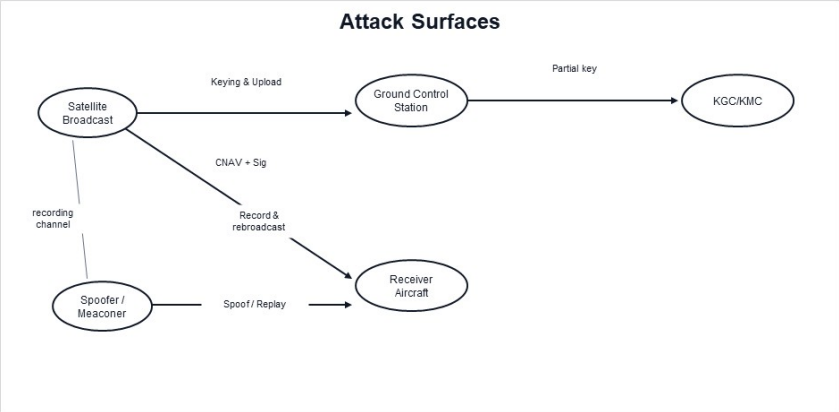

Figure 3. Main attack surfaces across satellite-ground-receiver paths

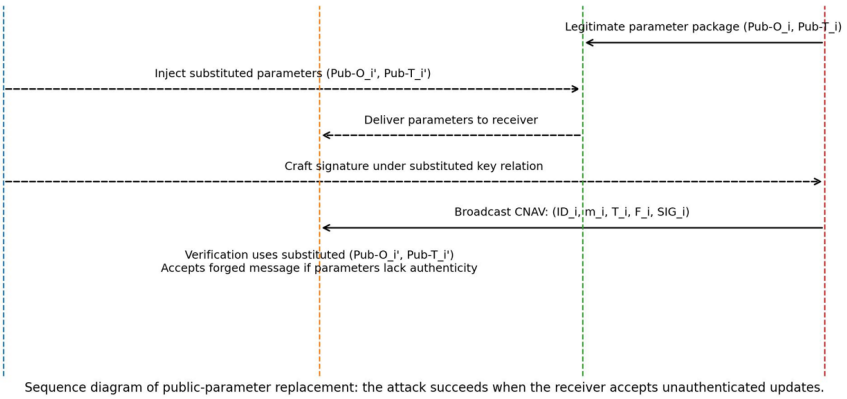

Figure 4. Sequence diagram of public-parameter replacement leading to forgery when parameter updates are unauthenticated

These vulnerabilities do not contradict the original unforgeability reduction in the random-oracle model; they show that operational trust hinges on binding, parameter authenticity, and implementation correctness as much as on the hardness of discrete logarithms.

To address mixed-epoch batching concretely, the receiver applies Algorithm 1 (in the appendix) before batch verification. The flow uses aggregation as the primary fast path after structural admissibility checks, while individual verification is reserved for fallback diagnosis when a batch fails.

5 Proposed Improvements

We keep the pairing-free certificateless structure, then harden three practical failure points: trust anchors, nonce robustness, and epoch-consistent aggregation. In the design, aggregation is retained because it can still reduce cost when used as the primary batch-verification fast path after inexpensive structural checks, rather than as a redundant post-check after every individual signature has already been verified.

Trust anchor for public parameters. A certificateless design still needs an authenticated distribution path for public parameters that the receiver uses during verification. Our paper treats satellite public parameters as a signed package issued by the authority and distributed through operational channels that the receiver can authenticate using a pinned authority public key. This package binds the identity, validity interval, and parameter digest, which prevents public-parameter replacement without inflating per-message bandwidth. In aviation-style deployments, the receiver can provision the authority public key during certification or maintenance, similar to how other safety-critical roots of trust are installed. Concretely, the receiver stores the authority public key during provisioning or maintenance, verifies the signature on each parameter package before any CNAV batch is accepted, checks the validity interval, and then loads Pri-O_ID$_i$ and Pub-T_ID$_i$ into a trusted parameter store. Any unsigned, expired, or mismatched package is rejected before batch verification begins.

Deterministic nonce and implementation robustness. The signer must prevent nonce repetition across reboots and failure modes. We adopt deterministic nonce derivation seeded by device entropy and protected by monotone state. This blocks signing-key recovery from repeated nonces while keeping computation lightweight. Receivers

should validate curve points and reject unexpected identities before expensive verification.

Epoch-consistent aggregation. We redefine aggregation as the primary fast path rather than a redundant post-check. The receiver first performs inexpensive structural checks—canonical parsing, authenticated parameter lookup, point validation, epoch selection, and identity de-duplication—then executes one batch verification over the candidate set. Individual verification is reserved for fault isolation or logging only when the aggregate check fails. This preserves the computational rationale for aggregation while closing the cross-epoch mixing pitfall.

These steps keep TrustCNAV's structure recognizable while tying operational acceptance to enforceable checks and authenticated inputs.

6 Experimental Results and Evaluation

Table 1 summarizes replay handling, privacy, computation time, and the per-message authentication overhead for the three designs. The latency values should be read as reference software-side point estimates rather than deployment-certified hardware benchmarks. The reported latency figures should be read as representative software-side averages obtained from repeated runs under the same reference configuration.

Table 1. Protocol comparison summary with computation and communication overhead

Protocol	Replay Detection	Identity Privacy	Auth Fields Added (bytes)	0-bytes per CNAV (bytes)	Signing Time (ms)	Mean Receiverside Authentication Latency L_{avg} (ms)	Compromise Containment
Baseline (Unauthenticated CNAV)	No	None	0	0	0.00	0.15	None
Original TrustCNAV	Windowbased	Moderate	65 (F_i + SIG_i)	65	2.10	3.50	Weak vs nonce fault/key leakage
Improved (Proposed)	Epochinvariant + cache	Improved	67 (add EpochID)	67	2.25	4.05	Stronger, limited blast radius

We evaluate three designs: (a) baseline CNAV without cryptographic authentication, (b) original TrustCNAV, and (c) the improved protocol with authenticated parameter packaging, deterministic nonces, and epoch-invariant batch filtering. The evaluation should be read as a reference software study rather than an embedded hardware certification benchmark. To make the modeling assumptions explicit, Table 2 lists the configuration used in the software-side comparison. In this paper, the values in Table 1 are interpreted as representative averages from a software-based reference implementation study, and Table 2 makes the implementation environment, cryptographic instantiation, and run-to-run stability assumptions explicit.

Metrics follow:

$$\begin{aligned} L_{\text{avg}}(n) &= \frac{1}{n}\sum_{k=1}^{n}(t_{\text{end},k} - t_{\text{start},k}) \\ S_{\text{auth}} &= \frac{N_{\text{succ}}}{N_{\text{total}}} \\ D_{\text{rep}} &= \frac{N_{\text{rep\_detected}}}{N_{\text{rep\_injected}}} \\ C_{\text{res}} &= \frac{N_{\text{sessions\_secure}}}{N_{\text{sessions\_total}}} \end{aligned} \tag{21}$$

We also add an overhead metric tailored to limited-bitrate navigation messages:

$$O_{\text{bytes}} = \text{additional authentication bytes per CNAV message} \tag{22}$$

Because the present study compares protocol variants under the same reference workload, these metrics should be interpreted as software-side comparative indicators rather than certified field-performance guarantees.

Communication overhead accounting. TrustCNAV transmits an elliptic-curve point F_i and a scalar SIG_i as its authentication data. Using compressed point encoding, the footprint is:

$$\begin{aligned} O_{\mathrm{bytes}} &= 33 \text{ bytes for } F_i + 32 \text{ bytes for } \mathrm{SIG}_i \\ O_{\mathrm{bytes}} &= 65 \text{ bytes} \end{aligned} \tag{23}$$

Our improved design binds an explicit epoch identifier to defeat mixed-epoch aggregation and replay acceptance. When the epoch identifier is transmitted, it can be encoded in 2 bytes:

$$O_{\mathrm{bytes}} = 65 \text{ bytes} + 2 \text{ bytes} = 67 \text{ bytes} \tag{24}$$

Table 2. Reference configuration of the software-side evaluation and basic stability check

Parameter	Reference Configuration Used in the Software-Side Study
Evaluation scope	Software-based reference implementation study of receiver-side authentication logic only; radio frequency (RF) acquisition, tracking, and navigation filtering excluded.
Implementation environment	Reference software implementation executed in Python 3.11 on a desktop-class x86-64 environment; no embedded prototype or RF front-end was used in this study
Cryptographic group model	NIST P-256 (secp256r1) prime-order elliptic curve group with compressed point encoding (33-byte point, 32-byte scalar).
Hash instantiation	SHA-256-derived fixed-length 256-bit outputs for H_1, H_2, and H_3 in the reference model.
Epoch and visibility	N_{sat} = 24 satellites, 1 Hz update per satellite, visibility-driven batch size n = 10 to 18.
Freshness parameters	Reference freshness window ΔT = 2 s; epoch duration = 1 s; minimum accepted batch size $n_{\min}$ = 4.
Attacker actions	Replay, unauthenticated parameter substitution, nonce-fault exploitation, and mixed-epoch buffering.
Nonce-fault model	Fault event after reboot may repeat the immediately previous nonce with probability 0.01.
Replay cache key	$(\mathrm{ID}_i, e^*, \mathrm{Hash}(m_i, F_i))$ after accepted batch verification.
Repeated runs	Table 1 values are representative averages over repeated software-side runs under the same reference workload; only minor run-to-run variation was observed, and the qualitative ranking of the three designs did not change.
Interpretation	Illustrative software-side comparison of design variants, not a deployment-ready benchmark for embedded rail, aviation, or unmanned aerial vehicle (UAV) receivers.

The parameter package used to authenticate Pri-O_ID$_i$ and Pub-T_ID$_i$ is distributed out-of-band and does not inflate per-message CNAV size, which addresses the bandwidth concern for continuous broadcasts.

Latency and security outcomes. Under the reference software model, baseline CNAV yields high S_{auth} under benign conditions but fails under injection because it lacks a cryptographic gate, and D_{rep} stays near zero. Original TrustCNAV increases D_{rep} when the timestamp window is tight, yet parameter substitution causes false acceptance if the receiver does not authenticate updates. Under a nonce-fault probability of 0.01 after a reboot, an attacker can recover the effective signing key after observing two signatures with repeated public nonce points, reducing C_{res}. The improved protocol prevents these failures in the modeled workloads by rejecting unauthenticated parameters, preventing nonce repetition, and enforcing epoch-consistent batching. The resulting authentication budget remains within the 1 Hz epoch window assumed in this study, but the accumulated 40.5–72.9 ms software-side cost for n = 10 to 18 authenticated satellites may still be material for embedded receivers with tighter CPU, power, or real-time constraints. We therefore treat the reported values as reference software timings rather than deployment-ready guarantees for rail, aviation, or autonomous-vehicle platforms. In the representative railway-positioning scenario emphasized in this paper, a failed authenticated batch would mean that the affected GNSS message set is withheld from the train-location solution or handled in a degraded cross-checked mode rather than being silently accepted. This interpretation underscores that the reported timing values describe a protected message-screening stage in software, not a universal claim of end-to-end real-time suitability across all transport platforms.

Bounded symbolic trace-exploration results. The symbolic exploration procedure reports accepting attack traces for the original TrustCNAV variant, matching the vulnerabilities described in Section 4.2, and reports no accepting

traces for the improved variant under the same modeled capabilities and search bounds. This analysis is deliberately limited: it checks only the modeled classes of nonce reuse, unauthenticated parameter replacement, replay handling, and mixed-epoch buffering. It should therefore be read as bounded symbolic evidence for the redesigned acceptance logic rather than as a general proof of protocol security.

Table 1 and Table 2 summarize the comparative overhead and reference evaluation assumptions used in the manuscript.

7 Conclusions

This paper presents an independent security reassessment of TrustCNAV by adding communication-overhead accounting, explicit trust-anchor discussion, transport-scenario interpretation, and a bounded symbolic trace-exploration layer. We showed that nonce faults and unauthenticated public-parameter updates can enable practical forgeries, and that weak epoch policies can undermine aggregation safety. The improved variant preserves pairing-free verification while adding authenticated parameter packaging, deterministic nonce generation, and epoch-invariant batch filtering. In the reference software study, these changes improve replay handling and compromise containment at the cost of additional verification latency. Important limitations remain: the present work does not defeat distance-decreasing attacks, the symbolic exploration is bounded and attack-class specific, and the reported timings are not hardware-level benchmarks for embedded transport receivers. Future work should therefore combine mechanized verification, authenticated-ranging or cross-sensor defenses, and platform-specific benchmarking on realistic rail, aviation, maritime, and autonomous-vehicle hardware. The practical deployment discussion is intentionally centered on GNSS-based railway positioning as a representative transport case, while broader applicability to aviation, maritime, or road platforms would still require platform-specific validation.

Funding

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (Grand No.: NRF-RS-2023-00237287).

Data Availability

The data used to support the research findings are available from the corresponding author upon request.

Conflicts of Interest

The author declares no conflict of interest.

References

[1] Z. Wu, Y. Bai, Y. Zhang, L. Liu, and M. Yue, "TrustCNAV: Certificateless aggregate authentication of civil navigation messages in GNSS," *Comput. Secur.*, vol. 148, p. 104172, 2025. https://doi.org/10.1016/j.cose.2024.104172

[2] T. E. Humphreys, "Detection strategy for cryptographic GNSS anti-spoofing," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 2, pp. 1073–1090, 2013. https://doi.org/10.1109/TAES.2013.6494400

[3] K. Zhang, E. G. Larsson, and P. Papadimitratos, "Protecting GNSS open service navigation message authentication against distance-decreasing attacks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 2, pp. 1224–1240, 2022. https://doi.org/10.1109/TAES.2021.3122512

[4] S. Wang, J. Liu, B. Cai, J. Wang, D. Lu, and W. Jiang, "Anti-spoofing performance analysis of typical GNSS-based railway train positioning schemes," *High-Speed Railw.*, vol. 3, no. 1, pp. 37–43, 2025. https://doi.org/10.1016/j.hspr.2025.01.005

[5] M. L. Psiaki and T. E. Humphreys, "GNSS spoofing and detection," *Proc. IEEE*, vol. 104, no. 6, pp. 1258–1270, 2016. https://doi.org/10.1109/JPROC.2016.2526658

[6] Z. Wu, Y. Zhang, Y. Yang, C. Liang, and R. Liu, "Spoofing and anti-spoofing technologies of global navigation satellite system: A survey," *IEEE Access*, vol. 8, pp. 165 444–165 496, 2020. https://doi.org/10.1109/ACCESS.2020.3022294

[7] J. N. Gross, C. Kilic, and T. E. Humphreys, "Maximum-likelihood power-distortion monitoring for GNSS-signal authentication," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 1, pp. 469–475, 2019. https://doi.org/10.1109/TAES.2018.2848318

[8] D. Margaria, B. Motella, M. Anghileri, J. J. Floch, I. Fernandez-Hernandez, and M. Paonni, "Signal structure-based authentication for civil GNSSs: Recent solutions and perspectives," *IEEE Signal Process. Mag.*, vol. 34, no. 5, pp. 27–37, 2017. https://doi.org/10.1109/MSP.2017.2715898

[9] M. Nicola, B. Motella, M. Pini, and E. Falletti, "Galileo OSNMA public observation phase: Signal testing and validation," *IEEE Access*, vol. 10, pp. 27 960–27 969, 2022. https://doi.org/10.1109/ACCESS.2022.3157337

[10] G. Seco-Granados, D. Gómez-Casco, J. A. López-Salcedo, and I. Fernández-Hernández, "Detection of replay attacks to GNSS based on partial correlations and authentication data unpredictability," *GPS Solutions*, vol. 25, pp. 1–15, 2021. https://doi.org/10.1007/s10291-020-01049-z

[11] B. Motella, M. Nicola, and S. Damy, "Enhanced GNSS authentication based on the joint CHIMERA/OSNMA scheme," *IEEE Access*, vol. 9, pp. 121 570–121 582, 2021. https://doi.org/10.1109/ACCESS.2021.3107871

[12] C. O'Driscoll and I. Fernandez-Hernandez, "Mapping bit to symbol unpredictability with application to Galileo open service navigation message authentication," *Navigation*, vol. 69, no. 2, 2022. https://doi.org/10.33012/navi.519

[13] I. Fernandez-Hernandez, T. Ashur, and V. Rijmen, "Analysis and recommendations for MAC and key lengths in delayed disclosure GNSS authentication protocols," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 3, pp. 1827–1839, 2021. https://doi.org/10.1109/TAES.2021.3053129

[14] I. Fernandez-Hernandez, J. Winkel, C. O'Driscoll, S. Cancela, R. Terris-Gallego, J. A. López-Salcedo, G. Seco-Granados, A. D. Chiara, C. Sarto, D. Blonski *et al.*, "Semi-assisted signal authentication for Galileo: Proof of concept and results," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 4, pp. 4393–4404, 2023. https://doi.org/10.1109/TAES.2023.3243587

[15] K. Ghorbani, N. Orouji, and M. R. Mosavi, "Navigation message authentication based on one-way hash chain to mitigate spoofing attacks for GPS L1," *Wirel. Pers. Commun.*, vol. 113, no. 4, pp. 1743–1754, 2020. https://doi.org/10.1007/s11277-020-07289-z

[16] Z. Wu, R. Liu, and H. Cao, "ECDSA-based message authentication scheme for BeiDou-II navigation satellite system," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 4, pp. 1666–1682, 2019. https://doi.org/10.1109/TAES.2018.2874151

[17] L. Crosara, F. Ardizzon, S. Tomasin, and N. Laurenti, "Worst-case spoofing attack and robust countermeasure in satellite navigation systems," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 2039–2050, 2024. https://doi.org/10.1109/TIFS.2023.3340061

[18] F. Ardizzon, L. Crosara, S. Tomasin, and N. Laurenti, "On mixing authenticated and non-authenticated signals against GNSS spoofing," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 4480–4493, 2024. https://doi.org/10.1109/TIFS.2024.3381473

[19] European Union, "Galileo Open Service Navigation Message Authentication (OSNMA) Service Definition Document," 2025, Luxembourg: Publications Office of the European Union. https://www.gsc-europa.eu/galileo/services/galileo-open-service-navigation-message-authentication-osnma

[20] A. Rusu-Casandra and E. S. Lohan, "Experimental assessment of OSNMA-enabled GNSS positioning in interference-affected RF environments," *Sensors*, vol. 25, no. 3, p. 729, 2025. https://doi.org/10.3390/s25030729

[21] I. Ali and F. Li, "An efficient conditional privacy-preserving authentication scheme for vehicle-to-infrastructure communication in VANETs," *Veh. Commun.*, vol. 22, p. 100228, 2020. https://doi.org/10.1016/j.vehcom.2019.100228

[22] W. Yang, S. Wang, and Y. Mu, "An enhanced certificateless aggregate signature without pairings for e-healthcare system," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 5000–5008, 2021. https://doi.org/10.1109/JIOT.2020.3034307

[23] M. Yang, J. Chen, Y. Chen, R. Ma, and S. Kumar, "Strong key-insulated secure and energy-aware certificateless authentication scheme for VANETs," *Comput. Electr. Eng.*, vol. 95, p. 107417, 2021. https://doi.org/10.1016/j.compeleceng.2021.107417

[24] Z. Wu, C. Liang, and Y. Zhang, "Blockchain-based authentication of GNSS civil navigation message," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 4, pp. 4380–4392, 2023. https://doi.org/10.1109/TAES.2023.3241041

[25] B. Blanchet, "An efficient cryptographic protocol verifier based on Prolog rules," in *Proceedings of 14th IEEE Computer Security Foundations Workshop, 2001*, Cape Breton, Canada, 2001, pp. 82–96. https://doi.org/10.1109/CSFW.2001.930138

[26] S. Meier, B. Schmidt, C. Cremers, and D. Basin, "The TAMARIN prover for the symbolic analysis of security protocols," in *Computer Aided Verification. CAV 2013. Lecture Notes in Computer Science (vol. 8044)*. Berlin, Heidelberg: Springer, 2013, pp. 696–701. https://doi.org/10.1007/978-3-642-39799-8_48

Appendix

Algorithm 1 Epoch-invariant candidate filtering and batch verification

Input:

- Buffer B of received CNAV messages $M_i = (\mathrm{ID}_i, m_i, T_i, F_i, \mathrm{SIG}_i)$.

Output:

- Accept/Reject decision.
- Accepted set S.

Step 1: Message Parsing and Epoch Extraction

(a) For each $M_i \in B$, parse the CNAV fields using a canonical encoding; reject the message if parsing is ambiguous.

(b) Compute $\mathrm{epoch\_id}_i = \mathrm{Epoch}(T_i)$ and store $(\mathrm{epoch\_id}_i, \mathrm{ID}_i, M_i)$.

Step 2: Target Epoch Selection

(a) Select the target epoch e^* with the largest count among the $\mathrm{epoch\_id}_i$ values satisfying

$$|T_{\mathrm{local}} - T_i| \le \Delta T.$$

Step 3: Candidate and Seen Set Initialization

(a) Initialize $\mathrm{Candidate} = \emptyset$ and $\mathrm{Seen} = \emptyset$.

Step 4: Epoch-Consistent Candidate Filtering

(a) For each message M_i with $\mathrm{epoch\_id}_i = e^*$, skip it if ID_i is unexpected or already appears in Seen; otherwise, add ID_i to Seen.

(b) Validate the points $\mathrm{Pub\text{-}O\_ID}_i$, $\mathrm{Pub\text{-}T\_ID}_i$, and F_i by checking curve membership and correct order.

(c) Recompute

$$h_{1,i} = \mathrm{H}_1(\mathrm{ID}_i, \mathrm{Pub\text{-}O\_ID}_i, \mathrm{Mas\text{-}Pub})$$

and

$$h_{3,i} = \mathrm{H}_3(\mathrm{ID}_i, m_i, \mathrm{Pub\text{-}Key\_ID}_i, F_i, T_i)\,.$$

(d) Require that the parameter package for ID_i is authenticated, then add M_i to $\mathrm{Candidate}$.

Step 5: Minimum Candidate Threshold Check

(a) If $|\mathrm{Candidate}| < n_{\min}$, reject.

Step 6: Batch Verification over Candidate Set

(a) Compute

$$F_{\mathrm{sum}} = \sum_i F_i$$

and

$$\mathrm{SIG}_{\mathrm{sum}} = \left(\sum_i \mathrm{SIG}_i\right) \bmod p.$$

(b) Accept only if

$$\mathrm{SIG}_{\mathrm{sum}} \cdot P = F_{\mathrm{sum}} + \sum_i h_{3,i} \cdot (\mathrm{Pub\text{-}T\_ID}_i + h_{1,i} \cdot \mathrm{Mas\text{-}Pub})\,.$$

Step 7: Fault Isolation on Batch Failure

(a) If the batch check fails, optionally invoke individual verification on $\mathrm{Candidate}$ for fault isolation or logging, then reject the batch for navigation use.

Step 8: Acceptance and Replay Cache Update

(a) If the batch check succeeds, set

$$S = \mathrm{Candidate}$$

and update the replay cache with $(\mathrm{ID}_i, e^*, \mathrm{Hash}(m_i, F_i))$ for all accepted messages.