Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Replay-Resistant Admission and Permissioned Quorum Consensus in Connected Vehicle Networks: A Fail-Closed, Implementation-Grounded Security Study

Md Shahanur Islam Shagor

smshagor.ru@gmail.com

Voronezh State University of Forestry and Technologies https://orcid.org/0009-0003-7730-3202

Research Article

Keywords: V2X security, replay resistance, permissioned consensus, network admission, connected vehicles, adversarial validation

Posted Date: September 17th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11047001/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Abstract

Connected-vehicle security requires more than message authentication: a receiver must also decide whether an identity is admitted, whether a message is fresh, and whether a vote originates from a currently authorized validator. This article isolates those controls from the broader OmniGuard V2X platform and evaluates them as a permissioned coordination subsystem. The design combines fail-closed registry-gated vehicle admission, authenticated session-to-identity binding, bounded timestamp-and-nonce replay state, fixed epoch-scoped validator membership, HMAC-SHA-256 authentication of canonical vote records, and a configurable quorum whose default threshold is ceil(2N/3). The implementation was examined at OmniGuard V2X v4.0.0, commit edca843814da7567875c5248f0c67c8101f300d8. A deterministic consensus campaign defines 192 corrupted-vote trials spanning hash substitution, epoch mismatch, outsider voting, authentication-tag corruption, stale timestamps, invalid proposal identifiers, malformed hashes, and malformed tags, followed by two clean quorum-forming votes and one prohibited post-finalization vote. The recorded release validation reports zero unexpected campaign failures; consequently, 193 invalid or post-finalization cases are rejected while the two valid votes establish a two-of-three quorum. A separate replay-flood regression fills a 64-entry nonce cache, rejects overflow without evicting live nonces, and preserves rejection of previously accepted messages. Security analysis shows that the subsystem constrains outsider and replay-driven influence but is deliberately not a Byzantine-fault-tolerant protocol: an authorized coalition meeting quorum can still approve a malicious proposal. The result is a reproducible, assumption-explicit admission and coordination design for connected-vehicle research systems.

1 Introduction

Vehicle-to-everything (V2X) communication joins safety-relevant telemetry, cooperative awareness, infrastructure services, and software-defined control across a radio environment in which endpoints are mobile and messages can be observed, delayed, duplicated, or injected. Contemporary WAVE and European ITS security specifications define secured-message and certificate structures for this environment [1, 2], while ISO/SAE 21434 and UN Regulation No. 155 place cybersecurity within a lifecycle and governance context [3, 4]. These standards establish an important baseline, but a deployed or experimental V2X service still needs local rules for deciding who may enter a coordination domain, which messages are fresh, and whose votes are allowed to change shared state.

Authentication alone does not answer those questions. A node may prove possession of a credential yet still be outside the authorized membership of a particular vehicle fleet, roadside domain, or validator committee. Likewise, a cryptographically valid message can be unsafe if it is replayed after its original context, and a valid identity can be dangerous if the same vote is counted repeatedly or if an old epoch remains actionable after membership rotation. Surveys of V2X security repeatedly identify impersonation, replay, authorization, and trust management as persistent challenges [5–7]. Blockchain-oriented vehicular designs add a further distinction: permissioned participation may constrain who

contributes to a ledger, but it does not automatically provide Byzantine fault tolerance, Sybil resistance, or freshness [8–11].

This paper therefore treats admission, replay resistance, and quorum voting as one security plane rather than as independent checkboxes. The implementation studied is the permissioned synchronization and consensus subsystem of OmniGuard V2X v4.0.0 [18]. The subsystem binds network sessions to enrolled vehicle identities, rejects unconfigured admission by default, maintains bounded replay state that fails closed at saturation, restricts voting to an epoch-scoped validator set, authenticates canonical vote records, and finalizes a proposal only when a configured fraction of the full validator set has voted yes. The default quorum is two thirds, computed as $q(N)=\mathrm{ceil}(2N/3)$, where N is the configured validator count.

The contribution is intentionally narrower than the previously released broad OmniGuard V2X system preprint, which described an integrated cross-layer prototype spanning cryptography, identity, blockchain, learning, and operations. Here, the research question is whether a small permissioned V2X coordination layer can make its admission, replay, membership, freshness, and finalization boundaries explicit and reproducibly testable. No claim of production automotive certification, formally verified consensus, or Byzantine-fault-tolerant state-machine replication is made.

The article makes four contributions. First, it specifies a fail-closed admission path that separates identity authenticity from permission to join the network and from permission to vote. Second, it defines bounded replay protection that refuses new nonces when capacity is saturated rather than evicting live replay evidence. Third, it formalizes an epoch-scoped permissioned quorum with proposal-hash binding, validator binding, duplicate-vote rejection, proposal freshness checks, and immutable finalization. Fourth, it evaluates the design through deterministic mutation-based validation and focused regression tests while explicitly deriving the coalition and denial-of-service boundaries that the implementation does not solve.

2 Background and related work

2.1 V2X authentication, freshness, and admission

IEEE 1609.2–2025 specifies secured application and management message processing for WAVE devices [1], and ETSI TS 103 097 V2.2.1 profiles secure ITS data structures and certificate formats against the current IEEE base standard [2]. These standards motivate strong cryptographic identity and message protection, yet system-specific admission remains a policy decision. Recent surveys distinguish authenticity from the broader authorization and privacy problems of vehicular networking, and they continue to identify replay as a practical threat because old but valid traffic can be resent into a new operational context [5–7]. Timestamp checks reduce the acceptance window, while nonce tracking detects exact reuse; neither mechanism alone establishes who is authorized to join a private coordination domain.

The distinction matters for Sybil resistance. Douceur showed that identity multiplicity cannot generally be prevented in an open distributed system without an external source of identity control or resource constraint [13]. For a permissioned vehicle domain, a manufacturer, transport authority, certificate authority, fleet operator, or equivalent registry can provide such a governance boundary. This does not make the registry infallible; it changes the threat model from unconstrained identity creation to compromise or misconfiguration of an explicit admission authority.

2.2 Permissioned ledgers and quorum consensus

Permissioned blockchain systems such as Hyperledger Fabric separate membership from the mechanics of distributed transaction processing and allow consensus to be tailored to the deployment trust model [12]. Vehicular surveys similarly note that permissioned or consortium designs can reduce open-membership exposure while retaining auditable shared state [8–10]. Recent work has also combined blockchain and access control in operational networks, including JNSM's BBNAC architecture for permissioned network admission [17]. These systems reinforce the value of explicit membership, but their properties depend on the chosen consensus and identity mechanisms rather than on the word blockchain itself.

Classical Byzantine fault-tolerant protocols address a stronger problem. PBFT provides replicated-state-machine safety and liveness assumptions in the presence of Byzantine replicas [14], while HotStuff develops a responsive, linear-communication BFT design for partial synchrony [15]. The mechanism evaluated here is deliberately simpler: it authenticates votes from a fixed membership and applies a configurable quorum, but it has no PBFT-style prepare/commit phases, view-change protocol, or theorem that tolerates f arbitrary Byzantine validators. The distinction is essential because a quorum of authorized malicious validators can still force acceptance.

2.3 Message authentication and vote binding

The vote-authentication primitive is HMAC-SHA-256. HMAC is a keyed message-authentication construction whose security depends on the shared secret and the underlying hash function [16]. In this implementation, each validator is configured with secret material and an HMAC tag is computed over a domain-separated canonical representation of the proposal identifier, proposal hash, validator identity, Boolean vote, epoch, and proposal timestamp. Because the primitive is symmetric, the tag provides keyed integrity and origin authentication among holders of the configured secret; it should not be interpreted as public-key non-repudiation.

3 System model and threat model

3.1 Entities and trust boundaries

The model contains vehicle or roadside clients V, an admission registry R, a synchronization service S, a configured validator set A, and proposal state P. A client first proves knowledge of enrollment material associated with its vehicle identifier. A successful handshake creates a session bound to that identifier.

Only a bound session may submit a consensus vote, and the claimed voter identifier must match the bound vehicle identity. The same identity must also belong to the current validator set before a vote reaches the quorum engine. This creates three separate gates: enrollment, authenticated-session binding, and validator authorization.

The reference implementation supports certificate-authority, vehicle-manufacturer-registry, and transport-authority-registry admission policies. If no policy is configured, the identity layer defaults to DENY_UNCONFIGURED rather than OPEN_REGISTRATION. A global pre-shared-key path exists only as an explicit laboratory option. Proof-of-work and proof-of-stake policy names are recognized by metadata but are not implemented by the synchronization admission layer and therefore fail closed instead of silently degrading to open registration.

3.2 Adversary capabilities

The bounded adversary can observe and replay network traffic; submit stale messages; alter proposal fields; forge a claimed voter identifier; submit malformed proposal identifiers, hashes, or authentication tags; attempt to vote from a non-validator identity; duplicate or flip a previously cast vote; continue voting after finalization; and exploit finite replay-state capacity by sending many unique nonces. The adversary may control an enrolled node. The analysis therefore distinguishes outsiders from authorized insiders.

The model assumes that cryptographic secrets of uncompromised nodes remain secret, HMAC-SHA-256 is not forged, the admission registry is configured correctly, and clocks remain sufficiently synchronized for the configured freshness window. It does not assume that all authorized validators are honest. Instead, the security boundary of a malicious authorized coalition is derived explicitly from the quorum function.

Table 1
Threat model, controls, and explicit non-goals

Threat / condition	Implemented control	Security interpretation	Explicit non-goal
Unenrolled identity	Registry-gated lookup; deny when registry is absent or identity is missing	Blocks network admission without configured enrollment	Does not validate external registry governance
Exact message replay	Timestamp freshness plus bounded nonce cache	Previously accepted nonce remains non-reusable during its live window	Does not prevent denial-of-service attempts
Replay-cache saturation	Reject unknown nonce when cache is full; never evict live nonce	Preserves replay safety at the cost of availability	Does not guarantee service under nonce flooding
Outsider vote	Session identity must equal claimed voter and belong to validator set	Prevents non-members from contributing to quorum	Does not stop compromise of an authorized validator
Vote mutation	HMAC-SHA-256 over canonical vote fields	Binds proposal, epoch, identity, vote, and timestamp	Not a public-key signature or non-repudiation scheme
Duplicate / vote flip	One vote per validator per proposal	Prevents double counting and local equivocation	No cross-system slashing or accountability mechanism
Stale proposal / epoch	TTL check and exact epoch match	Rejects old context and rotated membership	Depends on clock and epoch management
Malicious authorized coalition	Quorum threshold q(N)=ceil(2N/3)	Makes required coalition size explicit	Not BFT; a quorum-sized malicious coalition can force acceptance

4 Replay-resistant admission architecture

4.1 Fail-closed identity admission

Admission is intentionally separate from identity authenticity. The metadata layer can report that a presented identity is cryptographically authentic while also reporting that identity creation is unconstrained under explicit open registration. For production-like registry modes, the synchronization service loads a per-identity enrollment mapping. If the mapping is present, a client must match an enrolled vehicle record. If the mapping is absent, the default path rejects admission with a registry-required error. This behavior prevents an operational misconfiguration from becoming an implicit open network.

4.2 Authenticated session binding

After admission, the synchronization server binds the authenticated connection state to the admitted vehicle identifier. Consensus traffic is accepted only from an authenticated session. A vote payload that claims a different voter identity from the bound vehicle is rejected before quorum processing, and a correctly bound identity is still rejected if it is absent from the validator set. This sequencing prevents a valid general-purpose vehicle credential from being treated as validator authority.

4.3 Bounded replay state

Replay protection uses bounded caches for handshake and session traffic; the server defaults to 4096 live entries per replay cache. A cache entry records a nonce until it expires under the message freshness policy. The unusual but security-relevant behavior occurs at saturation: if a nonce is not already present but all live slots are occupied, the membership check returns a replay-equivalent rejection. The implementation therefore never evicts a still-live nonce merely to admit a new one.

This policy is fail-closed with a deliberate availability tradeoff. Under a unique-nonce flood, legitimate fresh traffic can be rejected until old entries expire. Evicting live entries would improve short-term availability but could make previously accepted messages replayable. For safety-oriented coordination, the implementation chooses to preserve the replay invariant and expose saturation counters so the operator can detect the condition.

5 Permissioned quorum consensus

5.1 Validator membership and epoch state

The consensus engine is initialized with a non-empty mapping of validator identifiers to per-validator secret material. Each configured key must meet a minimum length requirement, and secret values are excluded from metadata output. Membership is fixed for an epoch. Epoch rotation requires a strictly increasing epoch identifier, replaces the validator mapping, and clears proposal state, so votes from removed members cannot carry authority into the new epoch.

5.2 Canonical vote and authentication tag

Each vote authenticates a domain-separated canonical tuple consisting of the protocol domain, epoch, proposal identifier, 64-hex-character proposal hash, voter identifier, Boolean vote value, and proposal timestamp. The HMAC tag is recomputed with the key associated with the claimed validator and compared before the vote is counted. Changing the proposal hash, epoch, identity, vote value, or timestamp without recomputing the tag under the correct validator secret causes rejection. The engine also validates field syntax independently so malformed identifiers, hashes, and tags do not reach state transition logic.

5.3 Quorum and finalization

Let N be the configured validator count, a the number of yes votes, and b the number of no votes. The default threshold is q(N)=ceil(2N/3). A proposal becomes ACCEPTED when a > = q(N). It becomes

REJECTED when the remaining uncast votes can no longer raise a to q(N). Otherwise it remains PENDING. The threshold is calculated from the full configured validator set rather than only from currently active voters, preventing a small subset from redefining quorum by participating alone.

Each validator may cast only one vote for a proposal. A repeated vote, including a vote flip, is rejected. Once a proposal is finalized, later votes are rejected rather than appended to an already decided tally. Proposal records are also bound to their first observed proposal hash, so a second validator cannot silently substitute another hash under the same proposal identifier.

6 Security analysis

6.1 Replay safety within the configured freshness window

Consider a message m that has been accepted with nonce n and timestamp t. While n remains live in the replay cache, any exact retransmission is rejected because n is present. If the cache reaches capacity, an unknown nonce is also rejected; therefore saturation does not remove n and cannot by itself make m replayable. Once n expires, the timestamp bound limits whether m can still pass freshness validation. Replay safety therefore depends jointly on nonce retention and timestamp skew, not on either mechanism in isolation.

6.2 Outsider exclusion and Sybil boundary

An outsider without an enrolled identity is rejected before a session is established. An enrolled but non-validator vehicle may communicate through permitted synchronization paths but is rejected from the vote path. Under a registry-backed policy this constrains the number of vote-capable identities to the externally enrolled set, addressing the open-identity condition described by Douceur [13]. However, the mechanism does not establish that the external authority issued identities correctly or that one physical vehicle cannot receive multiple authorized records. Sybil resistance is therefore conditional on registry governance rather than an intrinsic property of HMAC, blockchain, or the validator engine.

6.3 Duplicate, substitution, and stale-context resistance

A validator identity is used as the unique vote key within each proposal. After one vote is recorded, any second vote by the same validator is rejected, regardless of whether its Boolean value changes. The proposal hash is fixed on first observation; a later vote that authenticates a different hash is rejected as a hash mismatch. Votes with an epoch different from the engine epoch are rejected, and proposals outside the default 30-second freshness bound are rejected as stale. These controls prevent a set of common state-confusion paths without claiming a full distributed consensus proof.

6.4 Authorized-coalition boundary

The strongest limitation is straightforward. If an adversary controls q(N) authorized validator credentials in the same epoch, it can produce enough valid yes votes to finalize a proposal, even if the proposal is semantically malicious. This follows directly from the acceptance rule. For N = 3, q = 2; for N = 10, q = 7;

for N = 30, q = 20. The mechanism therefore provides permissioned quorum authorization, not Byzantine fault tolerance. PBFT and HotStuff solve a stronger replicated-state-machine problem under additional protocol assumptions [14, 15]. Calling the present mechanism BFT would overstate the evidence.

6.5 Denial-of-service tradeoff

Fail-closed saturation intentionally converts replay-state exhaustion into rejection rather than memory growth or unsafe eviction. This bounds memory and preserves the non-reuse invariant but creates a denial-of-service surface: an attacker capable of reaching the authenticated message path with many unique nonces may consume capacity and temporarily cause false rejection of legitimate traffic. Operational mitigation therefore requires rate limiting, segmentation, admission controls, cache telemetry, and appropriate expiration windows. The mechanism protects integrity under saturation; it does not guarantee availability under flooding.

7 Reproducible validation methodology

7.1 Repository snapshot

The study is grounded in the public OmniGuard V2X repository, release version 4.0.0 [18]. The inspected main-branch snapshot is commit edca843814da7567875c5248f0c67c8101f300d8, authored on 23 August 2026. The relevant modules are permissioned_consensus.py, replay_security.py, identity_security.py, sync_protocol.py, adversarial_validation.py, and the associated tests under tests/. The source tree also records the security-hardening evolution that introduced permissioned membership, replay-flood handling, deterministic adversarial validation, and later release validation.

7.2 Deterministic mutation campaign

The adversarial validator runner uses a fixed default seed (0x0A11CE27), 192 mutation iterations, and a bounded execution budget. Each iteration constructs a legitimate vote for a three-validator engine at epoch 9 and then corrupts one of eight security-critical fields or encodings: the proposal hash, epoch, voter identity, authentication tag, proposal timestamp, proposal identifier, hash encoding, or authentication-tag encoding. The expected outcome for every mutated vote is rejection. An accepted mutated vote is recorded as an unexpected security failure.

After the mutation loop, the campaign executes a clean state transition. VAL1 submits a valid yes vote and the proposal remains PENDING because the full three-member set yields q(3) = 2. VAL2 submits a valid yes vote and the proposal becomes ACCEPTED. A subsequent vote after finalization is expected to fail. Thus, the default campaign contains 195 consensus checks: 192 mutation trials, two valid quorum-forming votes, and one prohibited post-finalization vote.

7.3 Focused regression tests

The permissioned-consensus regression suite defines 11 focused tests covering full-set quorum calculation, duplicate-vote rejection, vote-flip rejection, unauthorized validators, invalid authentication

tags, proposal-hash substitution, stale proposals, wrong epochs, epoch rotation and key replacement, late votes after finalization, and secret-free metadata. Replay-flood tests separately exercise a 64-entry cache: 64 unique valid messages populate it, a 65th fresh message is rejected at saturation, and a replay of the earliest accepted message remains rejected. A second fixture verifies that expired entries restore capacity.

7.4 Evidence classification

The evaluation distinguishes implementation evidence from performance claims. The mutation and regression artifacts establish deterministic acceptance/rejection behavior for specified cases. They do not measure radio latency, fleet-scale throughput, consensus liveness under packet loss, hardware security module performance, or Byzantine agreement. Numerical claims in this paper are therefore limited to configuration values, case counts, quorum arithmetic, and recorded validation outcomes that are directly traceable to the repository.

7.5 Manuscript preparation assistance

OpenAI ChatGPT was used to assist with manuscript organization and language drafting from the author's software artifacts and research instructions. The author remains responsible for the research, verified the technical claims and references, and approved the final manuscript content. No generative model was used to create experimental measurements, execute the repository validation, or substitute for the recorded software evidence analyzed in this study.

8 Results

8.1 Deterministic consensus campaign

The release validation record reports zero unexpected failures for the deterministic adversarial campaigns. Under the default permissioned-consensus configuration, this means all 192 corrupted votes follow the rejection path, the first clean vote is accepted into state as PENDING, the second clean vote reaches the two-of-three threshold and finalizes the proposal as ACCEPTED, and the prohibited post-finalization vote is rejected. Aggregated over the permissioned-consensus campaign, 193 invalid or disallowed cases are rejected and two valid votes are accepted for processing.

Table 2
Default permissioned-consensus campaign outcome

Case group	Cases	Expected behavior	Recorded outcome interpretation
Corrupted signed-vote mutations	192	Reject each case	No mutated vote accepted in the zero-unexpected-failure release run
First clean vote (VAL1)	1	Accept vote; state PENDING	Accepted; quorum 1/2
Second clean vote (VAL2)	1	Accept vote; state ACCEPTED	Accepted; quorum 2/2 reached
Post-finalization vote	1	Reject	Rejected; finalized state not reopened
Total	195	193 rejected; 2 valid votes processed	Zero unexpected consensus-campaign failures

8.2 Replay-flood behavior

The bounded replay fixture demonstrates the intended safety-versus-availability behavior. With max_entries = 64, the first 64 unique, valid messages are admitted and stored. A 65th valid message carrying a previously unseen nonce is rejected because the cache is saturated. The cache length remains 64. Critically, replaying the first accepted message is still rejected, showing that saturation did not evict its live nonce. In the expiration fixture, 16 expired entries are purged and a fresh message is admitted, demonstrating that fail-closed saturation is temporary when freshness state ages out normally.

Table 3
Replay-state regression behavior

Condition	Configured capacity	Observed/required behavior	Security meaning
Fill with unique live nonces	64	64 valid messages admitted; cache size reaches 64	Memory use is bounded
Fresh nonce at saturation	64	Message rejected; cache remains 64	No live replay evidence evicted
Replay earliest accepted message	64	Message rejected	Saturation does not reopen a replay path
All entries expired	16	Expired state removed; next fresh message accepted	Capacity recovers through expiry rather than live eviction

8.3 Quorum boundary

Figure 2 shows the deterministic growth of the default acceptance threshold. The security interpretation is two-sided. Increasing N raises the number of authorized credentials an attacker must control to force

acceptance, but the system still has no Byzantine agreement guarantee once the attacker reaches q(N). Conversely, if N-q(N) + 1 validators withhold or cast no votes, a proposal may be unable to reach acceptance. Quorum configuration is therefore a policy tradeoff among authorization strength, expected availability, and administrative control of validator credentials.

Table 4
Security-property summary

Property	Supported?	Evidence / condition
Fail-closed admission when unconfigured	Yes	DENY_UNCONFIGURED default and registry-required synchronization path
Replay rejection within live freshness state	Yes	Timestamp plus bounded nonce tracking
Memory-bounded replay state	Yes	Fixed-capacity cache; saturation rejects unknown nonces
Outsider validator exclusion	Yes	Bound session identity plus fixed validator membership
Duplicate/equivocating vote rejection	Yes	Single vote per validator per proposal
Proposal-hash and epoch binding	Yes	Canonical authenticated vote plus engine checks
Immutable finalization state	Yes	Late vote rejected after proposal finalization
Open-network Sybil resistance	No	Requires external registry/admission governance
Byzantine fault tolerance	No	Quorum-sized authorized coalition can force acceptance
Availability under authenticated nonce flooding	No guarantee	Fail-closed saturation may reject legitimate traffic

9 Discussion

9.1 Why admission and consensus must be separated

One practical finding is that network admission and consensus authorization should not share a single binary trust decision. A vehicle may be legitimate for telemetry synchronization while not being a validator. Conversely, a validator credential should not be usable from a session bound to another vehicle identity. The implementation therefore composes admission and voting rather than treating one as a proxy for the other. This is consistent with permissioned-ledger architectures in which membership is an explicit subsystem [12, 17], but it is tailored to a vehicle-centric synchronization path rather than a general enterprise blockchain.

9.2 Fail-closed replay saturation as an explicit policy

Replay caches are often described only as sets of recently seen nonces. Finite capacity changes the security problem. An implementation that silently drops the oldest live nonce under load can turn a memory-protection mechanism into a replay vulnerability. The evaluated cache instead treats saturation as a reason to reject uncertain freshness. This behavior is attractive for research prototypes that prioritize integrity, but production deployment would need upstream rate controls and capacity engineering so an attacker cannot cheaply convert freshness protection into a prolonged outage.

9.3 Relationship to automotive standards

The design is complementary to, not a replacement for, IEEE 1609.2 and ETSI TS 103 097. Those specifications define interoperable secured-message and certificate structures [1, 2]. The present subsystem addresses application-level policy after or alongside credential validation: whether an identity belongs to a local domain, whether its message is fresh in the local service, and whether it is authorized to vote in the current epoch. The same distinction is useful under ISO/SAE 21434, where cybersecurity controls should be tied to explicit assets, threats, assumptions, and lifecycle responsibilities [3].

9.4 Comparison with stronger consensus protocols

The simplicity of a fixed 2/3 authenticated quorum is both an advantage and a limitation. It is easy to inspect, test, and integrate with a small research runtime. It also lacks the phases and fault model of PBFT or HotStuff [14, 15]. Replacing the engine with a BFT state-machine replication protocol would materially change the system: validator communication, view changes, leader behavior, replicated state, liveness assumptions, and cryptographic accountability would all require new implementation and evaluation. The current paper therefore avoids using BFT terminology for a mechanism that only enforces authenticated membership and quorum state transitions.

9.5 Relationship to prior OmniGuard V2X work

The broader OmniGuard V2X preprint reported the architecture of a multi-mechanism vehicle-security prototype. This article does not repeat its cross-layer evaluation, post-quantum, privacy, federated-learning, or ledger-performance claims. Instead, it focuses on controls introduced and hardened in the later repository line: default-deny admission, explicit validator membership, bounded replay behavior under saturation, epoch-scoped vote authentication, and deterministic mutation testing. The narrower scope is intended to make the evidence auditable and to avoid treating the existence of a blockchain or identity layer as proof of replay, Sybil, or Byzantine resistance.

10 Limitations and threats to validity

First, the consensus validation is deterministic software validation, not a fleet-scale network experiment. The three-validator mutation campaign is appropriate for checking state-machine rejection paths but cannot establish throughput, latency, packet-loss tolerance, or wide-area liveness. Second, the validator

tag is based on HMAC with configured symmetric secrets; compromise of a validator secret enables that identity to authenticate votes, and symmetric authentication does not provide public non-repudiation. Third, registry-backed identity admission constrains identity creation only as strongly as the external enrollment authority. A compromised or poorly governed registry can authorize malicious or duplicate identities.

Fourth, proposal freshness depends on clock synchronization. The default 30-second tolerance is an engineering setting, not a universal bound, and deployments with larger clock drift must calibrate it without creating an unnecessarily broad replay window. Fifth, fail-closed nonce-cache saturation sacrifices availability to preserve replay state. The tests establish that live nonces are not evicted; they do not quantify the traffic rate required to cause saturation in a real deployment. Sixth, the quorum rule does not tolerate a quorum-sized authorized malicious coalition and has no leader-election or view-change protocol. The results must therefore not be generalized to Byzantine consensus.

Finally, the study is implementation-grounded in one open-source codebase and release line. This improves traceability but limits external validity. Independent implementations, formal protocol modeling, packet-level V2X experimentation, HSM-backed validator keys, distributed clock-fault experiments, and repeated performance campaigns are needed before production or certification-oriented conclusions can be drawn.

11 Conclusion

This article presented an implementation-grounded security study of replay-resistant admission and permissioned quorum coordination for connected vehicle networks. The design makes four boundaries explicit: identity authenticity is not the same as admission; admission is not the same as validator authorization; message authentication is not the same as freshness; and permissioned quorum is not the same as Byzantine fault tolerance. The reference implementation defaults to deny when admission is unconfigured, binds authenticated sessions to enrolled vehicle identities, rejects stale and duplicate messages using bounded fail-closed replay state, limits voting to an epoch-scoped validator set, authenticates canonical vote records with HMAC-SHA-256, and finalizes proposals using a configurable quorum whose default is two thirds of the full membership.

Deterministic validation provides concrete evidence for those boundaries. In the default consensus campaign, 192 corrupted-vote mutations, two clean quorum-forming votes, and one prohibited post-finalization vote form a 195-case security exercise; the recorded release run reports zero unexpected failures. A separate replay-flood regression shows that cache saturation rejects new uncertain traffic without evicting live nonce evidence. The resulting subsystem is useful as a reproducible research control plane, provided its non-guarantees remain visible: availability can degrade under nonce flooding, identity scarcity depends on external governance, and a quorum-sized authorized coalition can force acceptance. Future work should replace symmetric validator tags with hardware-backed asymmetric

identities, evaluate a real BFT protocol when that stronger threat model is required, and measure the complete design over realistic V2X network conditions.

Declarations

Competing Interests

The author declares no competing financial or non-financial interests.

Ethics approval and consent to participate

Not applicable. This study evaluates software and security mechanisms and does not involve human participants, animals, or clinical data.

Consent for publication

Not applicable.

Funding

No external funding is declared for this study.

Author contributions

Md Shahanur Islam Shagor conceived the study, designed and implemented the software mechanisms, prepared the validation artifacts, analyzed the security boundaries, and reviewed and approved the manuscript.

Data and code availability

The source code, deterministic validation logic, regression tests, and implementation documentation used in this study are publicly available in the repository “Next-Gen-Blockchain-for-Vehicle-Security” (OmniGuard V2X), release v4.0.0. The manuscript is grounded in commit edca843814da7567875c5248f0c67c8101f300d8: https://github.com/smshagor-dev/Next-Gen-Blockchain-for-Vehicle-Security

References

1. IEEE: IEEE Std 1609.2–2025, IEEE Standard for Wireless Access in Vehicular Environments - Security Services for Application and Management Messages. IEEE Standards Association (2026).

2. ETSI: ETSI TS 103 097 V2.2.1 (2026-03), Intelligent Transport Systems (ITS); Security; Security header and certificate formats; Release 2. European Telecommunications Standards Institute (2026).
3. ISO/SAE: ISO/SAE 21434:2021, Road vehicles - Cybersecurity engineering. International Organization for Standardization / SAE International (2021). https://doi.org/10.4271/ISO/SAE21434
4. UNECE: UN Regulation No. 155 - Cyber security and cyber security management system. United Nations Economic Commission for Europe (2021).
5. Yoshizawa, T., Singelee, D., Muehlberg, J.T., Delbruel, S., Taherkordi, A., Hughes, D., Preneel, B.: A survey of security and privacy issues in V2X communication systems. ACM Comput. Surv. 55(9), Article 185, 1–36 (2023). https://doi.org/10.1145/3558052
6. Shawky, M.A., Shah, S.T., Abdrabou, M., Usman, M., Abbasi, Q.H., Flynn, D., Imran, M.A., Ansari, S., Taha, A.: How secure are our roads? An in-depth review of authentication in vehicular communications. Veh. Commun. 47, 100784 (2024). https://doi.org/10.1016/j.vehcom.2024.100784
7. Abbas, S., Talib, M.A., Ahmed, A., Khan, F., Ahmad, S., Kim, D.-H.: Blockchain-based authentication in Internet of Vehicles: a survey. Sensors 21(23), 7927 (2021). https://doi.org/10.3390/s21237927
8. Dwivedi, S.K., Amin, R., Das, A.K., Leung, M.T., Choo, K.-K.R., Vollala, S.: Blockchain-based vehicular ad-hoc networks: a comprehensive survey. Ad Hoc Netw. 137, 102980 (2022). https://doi.org/10.1016/j.adhoc.2022.102980
9. Alladi, T., Chamola, V., Sahu, N., Venkatesh, V., Goyal, A., Guizani, M.: A comprehensive survey on the applications of blockchain for securing vehicular networks. IEEE Commun. Surv. Tutor. 24(2), 1212–1239 (2022). https://doi.org/10.1109/COMST.2022.3160925
10. Meijers, J., Michalopoulos, P., Motepalli, S., Zhang, G., Zhang, S., Veneris, A., Jacobsen, H.A.: Blockchain for V2X: applications and architectures. IEEE Open J. Veh. Technol. 3, 193–209 (2022). https://doi.org/10.1109/OJVT.2022.3172709
11. Su, H., Dong, S., Zhang, T.: A hybrid blockchain-based privacy-preserving authentication scheme for vehicular ad hoc networks. IEEE Trans. Veh. Technol. 73(11), 17059–17072 (2024). https://doi.org/10.1109/TVT.2024.3424786
12. Androulaki, E., Barger, A., Bortnikov, V., et al.: Hyperledger Fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the Thirteenth EuroSys Conference, Article 30, 1–15 (2018). https://doi.org/10.1145/3190508.3190538
13. Douceur, J.R.: The Sybil attack. In: Druschel, P., Kaashoek, F., Rowstron, A. (eds.) Peer-to-Peer Systems, LNCS 2429, pp. 251–260. Springer, Berlin (2002). https://doi.org/10.1007/3-540-45748-8_24
14. Castro, M., Liskov, B.: Practical Byzantine fault tolerance. In: 3rd Symposium on Operating Systems Design and Implementation (OSDI 99), pp. 173–186. USENIX Association, New Orleans (1999).
15. Yin, M., Malkhi, D., Reiter, M.K., Gueta, G.G., Abraham, I.: HotStuff: BFT consensus with linearity and responsiveness. In: Proceedings of the 2019 ACM Symposium on Principles of Distributed

Computing, pp. 347–356 (2019). https://doi.org/10.1145/3293611.3331591

16. Krawczyk, H., Bellare, M., Canetti, R.: HMAC: Keyed-Hashing for Message Authentication. RFC 2104, IETF (1997). https://doi.org/10.17487/RFC2104
17. Korkuc, C., Genc, Y., Dogantuna, T., Afacan, E., Dener, M.: BBNAC: blockchain based network access control for IT/OT infrastructures. J. Netw. Syst. Manag. 34, 60 (2026). https://doi.org/10.1007/s10922-026-10033-w
18. Shagor, M.S.I.: OmniGuard V2X / Next-Gen Blockchain for Vehicle Security, version 4.0.0, commit edca843814da7567875c5248f0c67c8101f300d8. GitHub (2026). https://github.com/smshagor-dev/Next-Gen-Blockchain-for-Vehicle-Security

Figures

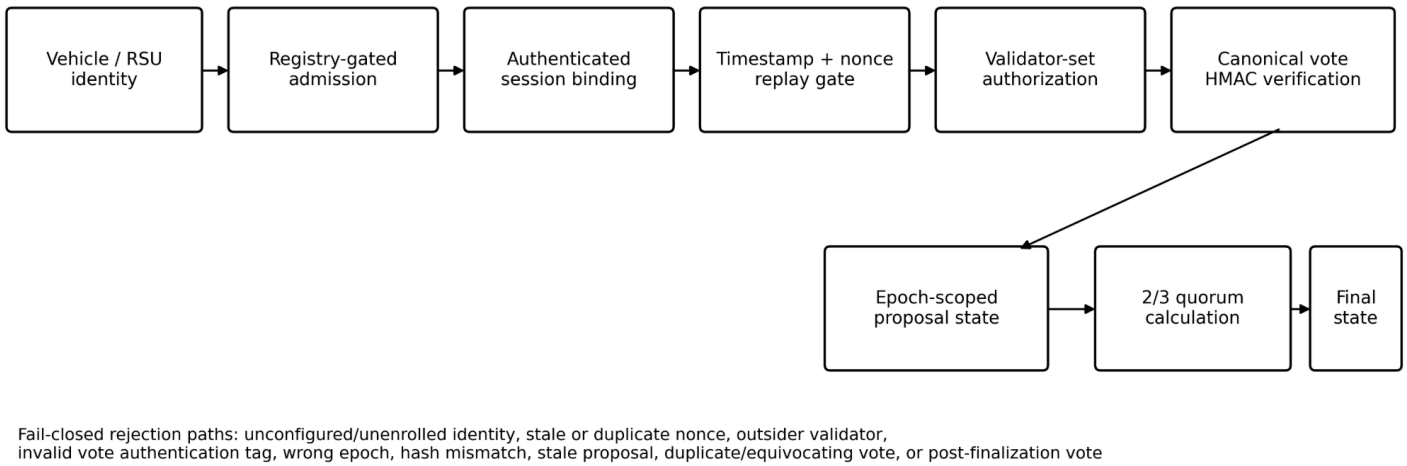


Figure 1

Admission, replay, validator-membership, vote-authentication, and quorum-finalization path

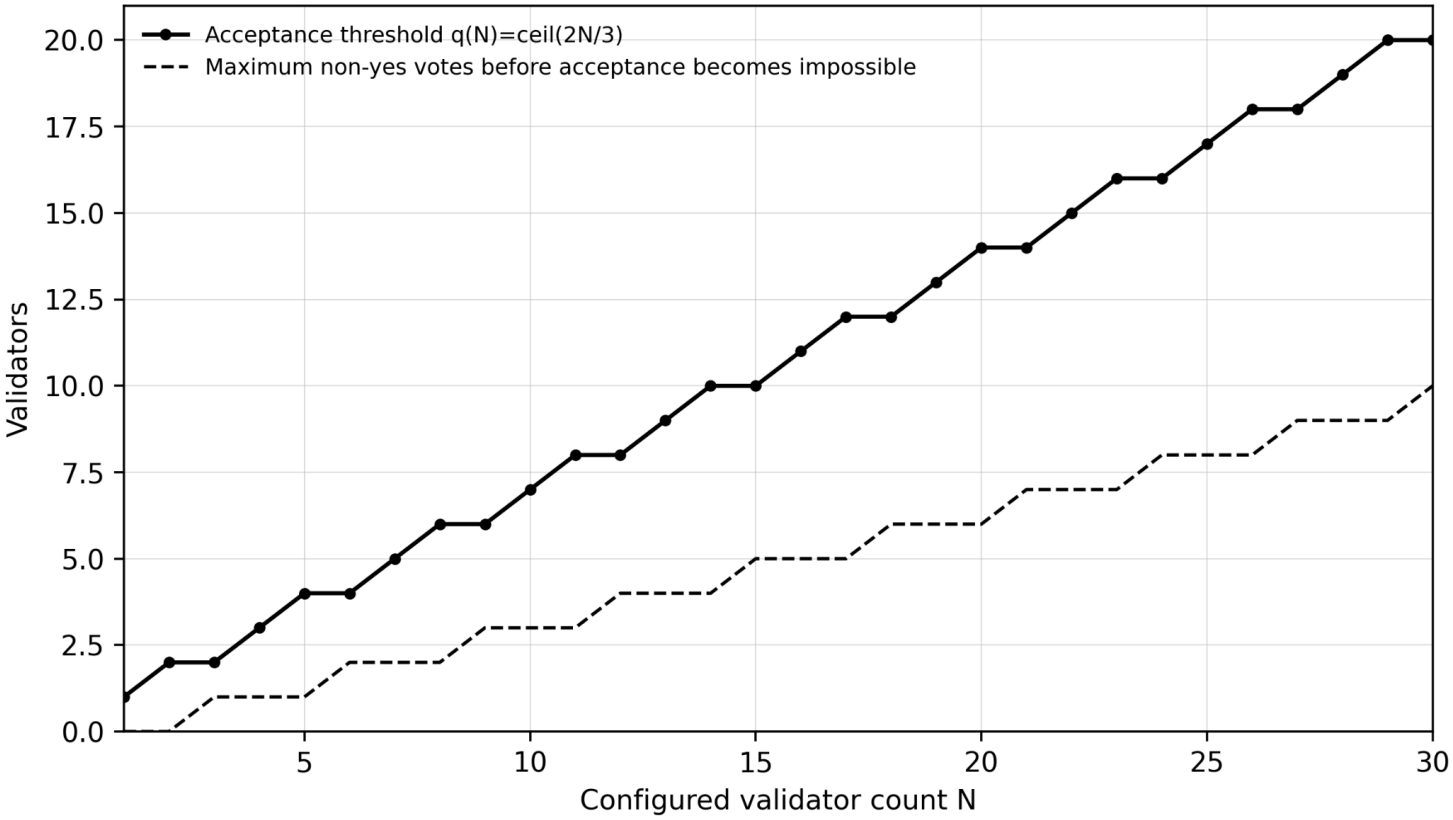


Figure 2

Default quorum threshold and corresponding non-yes margin as validator membership grows