

Secure-BSM: Blockchain-Assisted Collaborative Misbehavior Detection in Vehicular Networks

M M Imran

Department of Computer Science
Wayne State University
Detroit, MI 48201
Email: mmimran@wayne.edu

Yi Zhu

Department of Computer Science
Wayne State University
Detroit, MI 48201
Email: yzhu39@wayne.edu

Shiyong Lu

Department of Computer Science
Wayne State University
Detroit, MI 48201
Email: shiyong@wayne.edu

Abstract—Vehicle-to-Vehicle (V2V) communication relies on periodic Basic Safety Messages (BSMs) to support collision avoidance and cooperative driving, yet remains vulnerable to attacks such as False Data Injection (FDIA), spoofing, replay, Denial-of-Service (DoS), and Sybil attacks. Existing misbehavior detection approaches often exhibit significant limitations, as centralized authorities suffer from high revocation latency and infrastructure dependency, while standalone machine-learning and plausibility methods struggle with high false positive rates due to a lack of collaborative verification. We propose Secure-BSM, a blockchain-assisted architecture that reformulates vehicular misbehavior detection as a collaborative consensus problem to address these limitations. Shifting away from fixed-infrastructure models, Secure-BSM employs local attack detectors first, followed by decentralized validation where vehicles act as lightweight consensus validators. This dynamic approach utilizes a local Pocket Ledger for rapid consensus and a global Root Ledger for tamper-resistant revocation. By aggregating evidence via smart contracts and enforcing limits through Verifiable Rate-limit Tokens (VRTs), the system minimizes false positives and mitigates DoS attacks. Evaluation using SUMO traffic simulations across six attack types demonstrates that collaborative consensus improves detection precision by over 10% and reduces false positives by 77%–85% across various attack scenarios, while the VRT mechanism contains DoS flooding in under 0.2 seconds.

Index Terms—V2V Communication, Misbehavior Detection, Blockchain Consensus, BSM Security.

I. INTRODUCTION

Vehicle-to-Vehicle (V2V) communication is a foundational component of autonomous driving and cooperative intelligent transportation systems (C-ITS), enabling vehicles to exchange safety-critical information for collision avoidance, coordinated maneuvers, and traffic efficiency. In current standards, vehicles periodically broadcast Basic Safety Messages (BSMs), as defined under IEEE 1609.2 [1], containing kinematic state information such as position, velocity, acceleration, and heading at rates up to 10 Hz. This continuous information exchange underpins applications including forward collision warning, lane-change assistance, blind-spot detection, and cooperative adaptive cruise control. Despite these benefits, the broadcast and open-access nature of V2V wireless communication exposes vehicular networks to a broad range of cyberattacks [2]. Adversaries can inject falsified data, spoof legitimate identities, replay stale messages, flood channels to cause Denial-of-Service (DoS), or mount Sybil attacks using

multiple pseudonymous identities. Given the safety-critical role of BSMs, even a small number of malicious messages can propagate through vehicular platoons, triggering cascading traffic disruptions or accidents.

To secure V2V communication, existing deployments rely on the Security Credential Management System (SCMS) [3], which provides basic cryptographic authentication and pseudonym management through Certificate Authorities and centralized Misbehavior Authorities (MAs). MAs collect reports from vehicles and disseminate Certificate Revocation Lists (CRLs) via Road-Side Units (RSUs) or cellular infrastructure. However, SCMS exhibits fundamental limitations in adversarial and infrastructure-constrained environments [4]. First, the reliance on centralized entities creates heavy infrastructure dependency, rendering the system ineffective in rural or coverage-sparse regions. Second, the revocation latency is high; correlating reports and propagating CRLs can take minutes, leaving a dangerous vulnerability window. Third, standalone local detection methods utilized by vehicles are sensitive to sophisticated attack vectors such as gradual False Data Injection (FDIA) and Sybil attacks, leading to lower attack detection rates and high False Positive Rates (FPR) that risk wrongfully revoking honest vehicles. Finally, existing architectures lack effective countermeasures against high-rate DoS flooding at the BSM layer, where network saturation can occur before centralized revocation takes effect.

A substantial body of research has addressed vehicular misbehavior detection through plausibility checks [5], [6], machine learning classifiers [7], [8], and trust schemes [9], [10]. While these approaches identify various anomalous BSM patterns, they share a common limitation: detection is performed independently at each vehicle. Consequently, decisions are sensitive to local sensor noise, temporary occlusions, and environmental uncertainty, often leading to attack detection failure and high false positive rates. Furthermore, sophisticated adversaries can evade single-observer mechanisms by distributing malicious behavior across time and identities. A critical gap in existing solutions is a practical, infrastructure-independent mechanism that enables multiple vehicles to collaboratively validate suspicious behavior and reach agreement under adversarial conditions.

Blockchain technology offers decentralized consensus,

tamper-evident logging, and Sybil-resistant identity primitives that align naturally with vehicular security requirements. Prior research has explored blockchain for data sharing and trust management [11], [12]. However, most existing systems utilize blockchain as a passive backend operated by fixed infrastructure nodes, while vehicles act merely as lightweight clients [13], [14]. Such designs reintroduce infrastructure dependency and fail to support real-time, collaborative validation. Moreover, current blockchain-based proposals rarely address DoS attacks at the BSM layer, where high-rate message flooding can overwhelm both wireless bandwidth and on-board resources. Conventional rate-limiting typically relies on centralized enforcement, which conflicts with the dynamic nature of V2V environments.

This paper presents Secure-BSM, a blockchain-assisted architecture that reframes vehicular misbehavior detection as a collaborative consensus problem rather than an isolated local classification task. In contrast to prior systems, vehicles themselves act as blockchain validators, each running a lightweight Byzantine fault-tolerant node. Nearby vehicles dynamically form a validator set and execute an Istanbul Byzantine Fault Tolerant (IBFT 2.0) protocol to reach agreement on suspected misbehavior without relying on RSUs. The architecture comprises two complementary ledgers: a local *Pocket Ledger*, which enables rapid consensus among proximate vehicles for immediate validation, and a global *Root Ledger*, which supports long-term accountability, cross-regional correlation, and revocation management.

Within Secure-BSM, vehicles perform local detection using rule-based kinematic checks and temporal consistency analysis. When suspicious behavior is observed, evidence is submitted to the Pocket Ledger, where smart contracts aggregate reports from multiple independent observers. Misbehavior is confirmed only when a configurable reporting quorum is reached, significantly reducing false positives while improving robustness against coordinated attacks. To further address DoS threats, Secure-BSM introduces Verifiable Rate-limit Tokens (VRTs), a decentralized mechanism enforcing cryptographically verifiable transmission limits. VRT consumption is tracked on-chain, allowing receiving vehicles to locally verify whether a sender exceeds its permitted broadcast rate and reject excessive messages before processing.

We evaluate Secure-BSM using SUMO-based traffic simulations under six representative attack scenarios, including constant and gradual FDIA, spoofing, replay, DoS, and Sybil attacks. The results demonstrate that collaborative consensus improves detection precision by over 10% and reduces false positives significantly compared to local-only detection. Furthermore, the system maintains timely decision latencies suitable for cooperative driving safety margins, validating the feasibility of the vehicle-as-validator model on commodity automotive hardware.

The main contributions of this work are as follows:

- A novel vehicle-as-validator framework that eliminates reliance on infrastructure for local misbehavior detection.
- A two-tier ledger design supporting rapid local consensus

and global accountability, validated to reduce false positive rates.

- A new decentralized, cryptographically verifiable rate-limiting mechanism (VRT) for DoS mitigation within sub-second latency ($< 0.2s$).

The remainder of the paper is organized as follows: Section II reviews related work, Section III presents the Secure-BSM architecture, Section IV describes the implementation, Section V evaluates system performance, Section VI discusses limitations, and Section VII concludes the paper.

II. RELATED WORK

Research on misbehavior detection in vehicular networks has produced a broad spectrum of approaches, ranging from centralized revocation to decentralized, collaborative architectures.

A. Infrastructure-Centric Security

The foundational layer of vehicular security is the Security Credential Management System (SCMS), which relies on Public Key Infrastructure (PKI) to manage certificates and revocation [3]. In this model, Misbehavior Authorities (MAs) collect reports and distribute Certificate Revocation Lists (CRLs) via RSUs. However, recent performance analyses indicate that SCMS suffers from significant scalability and latency constraints [4]. The reliance on fixed infrastructure creates coverage gaps in rural areas, while the time required to aggregate reports and propagate CRLs can leave the network vulnerable to fast-acting attackers. These limitations have driven the shift toward local and collaborative detection mechanisms.

B. Local Detection Approaches

Early research primarily focused on plausibility- and physics-based detection, where vehicles locally validate received BSMs against physical and kinematic constraints. Representative works introduced acceptance range thresholds, speed and acceleration bounds, and displacement consistency checks to detect position falsification and replayed messages [5], [6]. Physical-layer plausibility approaches further incorporated signal characteristics, such as RSSI and camera-assisted verification, to strengthen detection under mobility uncertainty [15]. While these techniques are lightweight and interpretable, their effectiveness degrades in dynamic environments with sensor noise, sparse traffic density, or sophisticated attackers that remain within plausible bounds, leading to high false-positive rates and limited detection scope.

To address these limitations, subsequent studies integrated machine learning and deep learning into misbehavior detection pipelines. Supervised classifiers, including decision trees, random forests, and ensemble methods, were applied to engineered kinematic features [7], [16], while sequence models such as LSTMs and CNN-based architectures learned temporal dependencies in BSM streams [17], [18]. Systems such as DISMISS-BSM demonstrated improved detection of

gradual position spoofing by leveraging displacement compliance predictors [8]. Recent works have also explored Federated Learning to address privacy concerns while training these models [19], [20]. However, ML-based approaches face critical deployment hurdles. Beyond the need for extensive labeled data and frequent retraining to avoid overfitting, the computational complexity of these models often introduces inference latencies that exceed the strict millisecond-level processing deadlines required for safety-critical BSM exchange.

C. Collaborative and Trust-Based Systems

Beyond local detection, trust and reputation systems have been proposed to aggregate observations from multiple vehicles. These systems maintain dynamic trust scores for participants based on direct detections and indirect recommendations, often using probabilistic or fuzzy-logic models to update reputation over time [10], [16]. While collaborative trust models reduce reliance on single observers, most existing designs depend on centralized aggregation entities or cloud-based services to compute trust scores and manage revocation decisions. Such centralization introduces single points of failure, limited transparency, and revocation latencies that are incompatible with real-time vehicular safety requirements.

D. Blockchain-Assisted Security

In recent years, blockchain-assisted vehicular security has emerged to decentralize trust management. Prior research has explored blockchain for secure data sharing and privacy preservation [11]. Initially, blockchain was utilized as an immutable storage alternative to secure critical data. Early frameworks leveraged the ledger as a tamper-resistant repository for logging misbehavior reports, trust scores, and revocation lists, ensuring data integrity for post-hoc auditing without actively participating in real-time validation [12]. Subsequent works adopted Consortium Blockchain architectures. In these systems, Roadside Units (RSUs) or authorized edge nodes act as validators to manage the consensus, effectively retaining a reliance on fixed infrastructure to overcome the resource constraints of vehicles [14].

More recently, advanced integrations have been proposed to secure the collaborative training of detection models, such as Blockchain-Enabled Federated Learning (BC-FL) [21]. However, as noted in analyses of transparent architectures [13], these approaches largely remain dependent on RSU-based verification or offline storage paradigms. This design treats vehicles as lightweight clients, leaving a gap for fully decentralized, vehicle-centric real-time validation. Furthermore, while identity management is often addressed, existing blockchain frameworks lack decentralized mechanisms to enforce cryptographically verifiable rate limits against high-rate DoS flooding at the BSM layer, instead relying on simplistic local filtering that adaptive attackers can easily bypass.

In contrast to existing work, Secure-BSM advances the state of the art by treating misbehavior detection as a collaborative consensus problem rather than an isolated classification task or a post-hoc trust update. Unlike prior blockchain V2X systems

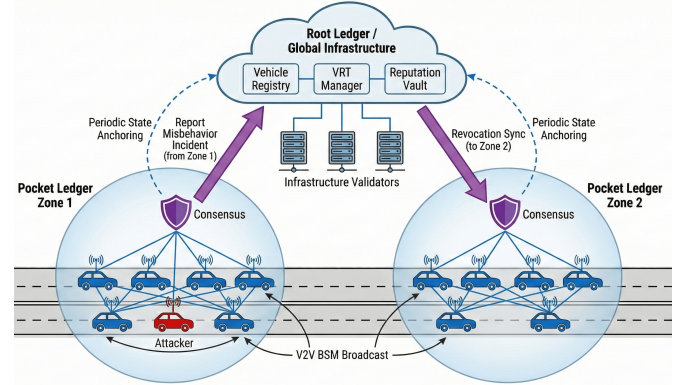


Fig. 1. Three layer architecture of Secure-BSM

where vehicles are passive clients, Secure-BSM adopts a vehicle-as-validator architecture in which vehicles themselves participate in Byzantine fault-tolerant consensus for misbehavior validation, eliminating infrastructure dependency for local decision-making. Moreover, the proposed Verifiable Rate-limit Token (VRT) mechanism introduces cryptographically enforceable, decentralized rate control, addressing the limitation of prior works by actively throttling adaptive adversaries. By integrating local detection, multi-vehicle consensus, and enforceable mitigation within a unified architecture, Secure-BSM complements and extends existing plausibility-based, ML-based, and blockchain-assisted approaches while explicitly addressing their limitations.

III. SYSTEM ARCHITECTURE

A. System Model and Design Overview

We model the vehicular network as a dynamic graph $G(t) = (\mathcal{V}, \mathcal{E}(t))$, where $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$ represents the set of vehicles and $\mathcal{E}(t)$ represents the set of active communication links at time t . Each vehicle v_i possesses a unique identity ID_i (pseudonym) and a public-private key pair (PK_i, SK_i) . Table I summarizes the key mathematical notations and parameters used throughout this paper.

The architecture is organized into three functionally distinct logical layers that operate at different spatial and temporal scales, as illustrated in Fig. 1:

- 1) **Local Layer (Detection Engine):** Each node v_i independently monitors neighbors $\mathcal{N}_i = \{v_j \mid (v_i, v_j) \in \mathcal{E}(t)\}$. The local detection function $f_{det} : \mathcal{M} \rightarrow \{0, 1\}$ maps incoming Basic Safety Messages (BSMs) to a binary

TABLE I
KEY MATHEMATICAL NOTATIONS AND PARAMETERS

Symbol	Definition & Default Value
f_{val}	Byzantine tolerance limit ($(\mathcal{V}_{val} - 1)/3$)
T_{rep}	Reporting quorum for misbehavior confirmation (≥ 3)
T_j	Trust Score (Range: [0, 100], Init: 50)
T_{val}	Validator Admission Threshold (> 90)
T_{revoke}	Global revocation threshold (Score < 20)
$\delta_{penalty}$	Trust penalty per incident (20 points)
C_{max}	VRT Token Bucket Capacity (5 tokens)
λ	VRT Refill Rate (10 tokens/s)
W_{dos}	Time window for rate limit detection (1.0 s)
r_{geo}	Pocket Ledger operational radius (150 m)

suspicion state (1: suspicious, 0: benign) by executing rule-based checks and cryptographic verifications.

- 2) **Consensus Layer (Pocket Ledger):** A subset of nodes $\mathcal{V}_{val} \subset \mathcal{V}$ forms a dynamic validator set for a specific geographic region. This layer functions as a regional Byzantine Fault Tolerant (BFT) blockchain where smart contracts aggregate detection reports from multiple observers. When a sufficient quorum of independent reporters concur on an accusation within a specific time window, the consensus promotes the report to a confirmed misbehavior incident.
- 3) **Global Layer (Root Ledger):** A persistent, network-wide ledger manages the lifecycle of identities $\mathcal{I}$ and enforces revocation decisions across disjoint geographic regions. This layer anchors evidence for long-term accountability. When a vehicle's trust score, managed by the Pocket Ledger, falls below a critical threshold, the system escalates the decision to the Root Ledger to issue a global revocation.

a) *Vehicle-as-Validator Architecture:* A fundamental innovation of this model is that vehicles themselves act as blockchain validators for the Pocket Ledger. Nodes dynamically join the validator set $\mathcal{V}_{val}$ upon entering a geographic region and gracefully depart upon exiting. This approach decouples real-time misbehavior consensus from Road-Side Units (RSUs), allowing the system to maintain security in infrastructure-sparse regions where centralized connectivity is intermittent or unavailable.

b) *Threat Model:* We assume a Byzantine adversary model where attackers control up to 30% of vehicles in a region (f nodes in a local validator set of size n). The system guarantees safety and liveness provided $n \geq 3f + 1$. Attackers may collude to execute the following vectors (see Fig. 2 for DoS scenario):

- *Data Injection:* Broadcasting false BSM data, such as constant positions or gradually drifting trajectories, using valid credentials.
- *Identity Attacks:*
 - *Sybil:* Leveraging multiple valid pseudonym certificates to simulate ghost traffic or influence consensus.
 - *Spoofing:* Broadcasting messages where the claimed ID does not cryptographically correspond to the signing key (Identity Consistency Failure).
- *Network Attacks:* Flooding communication channels (DoS) or replaying previously captured valid messages.

We assume honest vehicles possess accurate sensors and correctly implement detection protocols. Furthermore, we assume cryptographic primitives (ECDSA, Keccak-256) are secure and that the BFT consensus protocol maintains safety under the defined Byzantine conditions.

B. Local Misbehavior Detection and Mitigation

The local detection engine operates on a stream of BSMs $M_j = \{m_j^1, m_j^2, \dots\}$ received from neighbor v_j . Each message m_j^k contains a position p_j^k , velocity $\vec{v}_j^k$, acceleration $\vec{a}_j^k$,

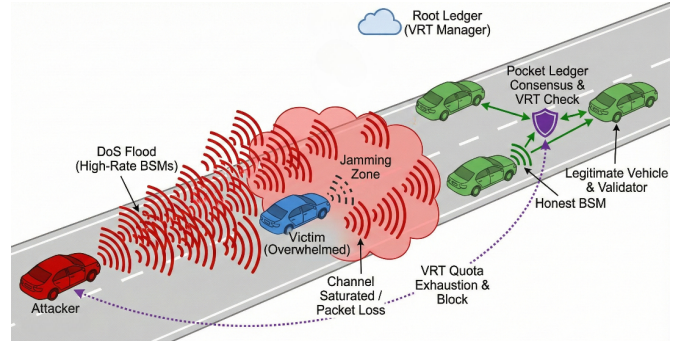


Fig. 2. High Level DoS Attack Scenario and Threat Model

yaw rate ω_j^k , timestamp t_j^k , and sequence number seq_j^k . The engine processes these inputs through specialized detectors to identify anomalies autonomously.

a) *Kinematic Consistency:* We verify that reported kinematics fall within physically achievable bounds [5]. We define a kinematic plausibility function $\Phi_{kin}(m_j^k, m_j^{k-1})$ based on Newtonian mechanics and vehicle dynamics. A message is flagged if it violates velocity, acceleration, or heading rate constraints:

$$\Phi_{kin} = \begin{cases} 1 & \text{if } \frac{\|p_j^k - p_j^{k-1}\|}{t_j^k - t_j^{k-1}} > v_{max} \\ 1 & \text{if } \|\vec{v}_j^k - \vec{v}_j^{k-1}\| > a_{max} \cdot (t_j^k - t_j^{k-1}) \\ 1 & \text{if } |\omega_j^k| > \omega_{max} \\ 0 & \text{otherwise} \end{cases}$$

where v_{max} , a_{max} , and ω_{max} (set to 45°/s) represent the maximum valid limits for velocity, acceleration, and heading rate respectively. This check detects blatant fabrications, such as instantaneous position jumps or impossible speeds.

b) *Displacement Compliance:* To detect Gradual False Data Injection Attacks (FDIA) where attackers incrementally drift position, we employ a Constant Velocity (CV) projection check. Let $\hat{p}_j^k$ be the predicted position based on the previous state vector $S_j^{k-1} = [p_j^{k-1}, \vec{v}_j^{k-1}]$ and time delta $\Delta t = t_j^k - t_j^{k-1}$. The displacement error ϵ_{disp} is defined as the Euclidean distance between the reported and predicted positions:

$$\epsilon_{disp} = \|p_j^k - \hat{p}_j^k\| \quad \text{where} \quad \hat{p}_j^k = p_j^{k-1} + \vec{v}_j^{k-1} \cdot \Delta t$$

A message is flagged if ϵ_{disp} exceeds the allowable physical tolerance δ_{err} (typically 2.5 meters), which accounts for measurement noise and minor maneuvers:

$$\Phi_{disp} = \mathbb{I}(\epsilon_{disp} > \delta_{err})$$

c) *Temporal Consistency:* To detect Replay Attacks involving cached or delayed messages, we enforce strict temporal constraints. A message m_j^k is accepted only if

$$|t_{now} - t_j^k| < \tau_{clock} \wedge t_j^k > t_j^{k-1} \wedge seq_j^k > seq_j^{k-1}$$

Here, τ_{clock} accounts for allowable GPS clock skew (e.g., 2 seconds). Messages violating monotonicity or sequence order are rejected, as are exact duplicates.

d) *Signature and Identity Verification:* Every BSM must carry a valid ECDSA signature σ_j^k verifiable by the sender's public key PK_j . The system verifies cryptographic integrity

and certificate status as:

$$\text{Verify}(PK_j, m_j^k, \sigma_j^k) = \text{True} \wedge PK_j \notin \mathcal{R}_{\text{revoked}}$$

The revocation status is checked against a local cache synchronized with the Root Ledger. Failures indicate Spoofing Attacks or the use of compromised credentials.

e) Sybil Detection: We define a Sybil Attack as the presence of a set of distinct identities $\mathcal{S} = \{ID_a, ID_b, \dots\}$ originating from the same physical trajectory. Using a spatial index for efficient radius queries, the detection condition is

$$\exists t, \forall ID_x, ID_y \in \mathcal{S} : \|p_x(t) - p_y(t)\| < R \wedge ID_x \neq ID_y$$

Here, R is the physical radius (e.g., 5 m).

f) DoS Detection and Mitigation: DoS defense operates on two levels: local detection and global cryptographic enforcement.

- **Local Detection:** A node v_j is flagged if its observed message rate λ_{obs} exceeds the protocol limit λ_{max} (typically 15 messages per 1.0 s), where $\lambda_{obs} = \frac{N_{msgs}}{W_{dos}} > \lambda_{max}$.
- **VRT Enforcement:** To prevent network congestion, the system employs Verifiable Rate-limit Tokens (VRT) maintained on the ledger, as detailed in Section III-E.

g) Reporting Mechanism: When any detector flags misbehavior, the vehicle constructs a structured `MisbehaviorReport` containing the accused pseudonym, attack type, timestamp, and an evidence hash. This report is signed and submitted as a transaction to the Pocket Ledger for collaborative consensus.

C. Pocket Ledger (Consensus Layer)

The Pocket Ledger constitutes a regionally scoped, dynamic permissioned blockchain that facilitates collaborative misbehavior validation. Unlike static infrastructure-based ledgers, the Pocket Ledger is maintained by the vehicles themselves, operating under an IBFT 2.0 (Istanbul Byzantine Fault Tolerant) consensus protocol [22]. The complete distributed detection and consensus procedure is outlined in Algorithm 1.

a) Dynamic Validator Set Formulation: We define the validator set $\mathcal{V}_{val}^{(k)}(t)$ for a geographic region R_k as a dynamic subset of vehicles satisfying spatial and reputational constraints. As vehicles enter a region, they dynamically join the set; as they exit or time out, they are gracefully removed. A vehicle v_i continuously monitors its spatial eligibility using the standard GNSS/GPS coordinates natively embedded in its IEEE 1609.2 BSMs. It is eligible to join if:

$$\text{Eligible}(v_i) \iff (\|p_i(t) - C_k\| \leq r_{geo}) \wedge (T_i \geq T_{val})$$

Here, C_k is the region centroid, r_{geo} is the operational radius (typically 150 m), and T_{val} is the validator admission threshold (e.g., Trust Score > 90).

The validator set evolves through a voting mechanism. A new node v_{new} is admitted if it receives approval votes from a supermajority of existing validators (Standard BFT Quorum):

$$\sum_{v_j \in \mathcal{V}_{val}^{(k)}} \text{Vote}(v_j, v_{new}) \geq 2f_{val} + 1$$

Algorithm 1 Distributed Misbehavior Detection & Consensus

Require: Incoming BSM stream $\mathcal{M}$, Validator Set $\mathcal{V}_{val}$

Ensure: Confirmed Misbehavior Event $D_{confirm}$

Phase 1: Local Detection (Run by every vehicle v_i)

- 1: **for** each new message m_j^k from neighbor v_j **do**
- 2: $\Phi_{kin} \leftarrow \text{CheckKinematics}(m_j^k, m_j^{k-1})$
- 3: $\Phi_{disp} \leftarrow \text{CheckDisplacement}(m_j^k, S_j^{k-1})$
- 4: **if** $\Phi_{kin} = 1$ **or** $\Phi_{disp} = 1$ **or** $\text{IsSybil}(m_j^k)$ **then**
- 5: **if** $\text{NotYetReported}(ID_j)$ **then**
- 6: $R \leftarrow \text{CreateReport}(ID_j, \text{Type}, \text{Proof})$
- 7: $\text{SubmitTransaction}(\text{PocketLedger}, R)$
- 8: **end if**
- 9: **end if**
- 10: **end for**

Phase 2: Collaborative Consensus (Run by Validators)

- 11: **while** System is Active **do**
- 12: **for all** suspect vehicle v_j with pending reports **do**
- 13: $\Omega_j \leftarrow \text{FetchPendingReports}(v_j)$
- 14: $\Psi(\Omega_j) \leftarrow \text{FilterUniqueReporters}(\Omega_j)$
- 15: **if** $|\Psi(\Omega_j)| \geq \tau_{rep}$ **then**
- 16: $D_{confirm}(v_j) \leftarrow \text{True}$
- 17: $T_j \leftarrow \max(0, T_j - \delta_{penalty})$
- 18: **if** $T_j < T_{revoke}$ **then**
- 19: $\text{EscalateToRootLedger}(ID_j, \text{Proof}_{\mathcal{V}_{val}})$
- 20: **end if**
- 21: **end if**
- 22: **end for**
- 23: **end while**

To prevent exponential communication overhead in high-density environments (e.g., urban intersections with > 40 vehicles), the smart contract enforces an active validator cap ($N_{max} \approx 10$). If the number of eligible vehicles exceeds this cap, the protocol selects a bounded subset to actively propose and validate blocks, while the remainder act as passive clients. This ensures the Pocket Ledger maintains sub-3-second latency regardless of physical traffic density.

Furthermore, to account for high-mobility churn, vehicles that exit the region mid-consensus are gracefully handled as standard offline nodes within the IBFT 2.0 Byzantine fault allowance (f). As long as a supermajority ($2f_{val} + 1$) of the active validators remains, consensus completes uninterrupted.

Sybil Resilience: By enforcing the strict reputation threshold (T_{val}), the system prevents Sybil identities from infiltrating the consensus layer. Since new or malicious identities cannot instantly generate a verifiable history of honest behavior to achieve T_{val} , they remain excluded from the validator set, preserving Byzantine fault tolerance even in the presence of identity spoofing.

b) Distributed Evidence Aggregation: The ledger acts as a deterministic state machine [23] aggregating local `MisbehaviorReport` transactions submitted by vehicles. The smart contract logic maintains a mapping of accusations and uses a time-window filter to group related reports. Let $\Omega_j(t)$ be the set of valid accusations against target vehicle v_j

within the current epoch. Each report $r \in \Omega_j$ is a tuple:

$$r = \langle ID_{reporter}, ID_{target}, Type, Evidence, t_{obs} \rangle$$

To prevent Sybil attacks on the reporting mechanism (where one malicious node submits multiple reports), the aggregation function Ψ filters for unique reporters. A report is valid only if the reporter has not previously voted on this specific incident within the time window W_{valid} (typically 15 seconds):

$$\Psi(\Omega_j) = \{ID_{reporter} \mid r \in \Omega_j \wedge t_{now} - r.t_{obs} \leq W_{valid}\}$$

Crucially, cryptographic evidence submitted via `MisbehaviorReport` transactions persists in the smart contract state for the duration of W_{valid} . This state persistence guarantees that stationary attackers attempting to continuously exploit high-speed passing traffic will rapidly accumulate the required reporting quorum and face revocation, even if the reporting vehicles subsequently exit the geographic region.

c) *Consensus Logic and Trust Evolution*: The system enforces a collaborative decision model to minimize false positives. A confirmed misbehavior event $D_{confirm}(v_j)$ is triggered if the cardinality of unique reporters exceeds the reporting quorum τ_{rep} .

$$|\Psi(\Omega_j)| \geq \tau_{rep} \implies D_{confirm}(v_j) = True$$

Distinct from the validator quorum ($2f_{val} + 1$) which secures ledger consistency, τ_{rep} defines the necessary witness count to validate a physical event (typically set to $\tau_{rep} \geq 3$ independent observers). Upon confirmation, the trust score T_j of the accused vehicle is updated. To prevent ‘‘Cold Start’’ Sybil attacks where adversaries generate fresh identities to infiltrate the validator set, scores initialize at a neutral value of 50. Honest behavior earns a reputation increment (+10 per hour), allowing legitimate long-term participants to reach the validator threshold ($T_{val} > 90$). Because of this slow accumulation rate, an attacker attempting a long-term Sybil farming attack must expend 16 hours of continuous, verifiable honest driving to farm the 4 identities required to subvert a standard 10-node Pocket Ledger ($f \geq 4$), rendering localized takeovers economically and practically irrational. Conversely, confirmed incidents impose immediate penalties:

$$T_j(t+1) = \max(0, T_j(t) - \delta_{penalty})$$

Here, $\delta_{penalty}$ is set to 20 points per incident. If $T_j(t+1) < T_{revoke}$ (critical threshold of 20), the Pocket Ledger initiates an escalation procedure to the Root Ledger for global revocation.

D. Root Ledger (Global Layer)

The Root Ledger serves as the persistent, globally consistent trust anchor for the network. Unlike the ephemeral, region-specific Pocket Ledgers, this layer maintains the state of the vehicular network indefinitely. It acts as the supreme authority for identity lifecycle management, global revocation enforcement, and long-term forensic auditability.

Fully decentralizing this global state across highly transient and resource-constrained vehicles would compromise the long-term persistence and high availability required for

global revocation routing. Instead, the Root Ledger functions as a Federated (Consortium) Blockchain. In a real-world deployment, these infrastructure validators would be operated by a consortium of trusted stakeholders, such as regional transportation authorities and automotive OEMs. This hybrid architecture effectively replaces the single-point-of-failure Misbehavior Authority (MA) of the traditional SCMS model. The core contribution is not the total eradication of infrastructure, but the removal of infrastructure from the latency-sensitive critical path; real-time hazard mitigation is decentralized to the Pocket Ledger at the network edge, while the federated Root Ledger is strictly utilized for asynchronous, long-term cryptographic accountability.

a) *Global Identity Registry*: The registry maintains the set of valid certificates $\mathcal{C}$ through automated smart contract execution. A certificate C_i binds a pseudonym ID_i to a public key PK_i and validity constraints:

$$C_i = \langle ID_i, PK_i, t_{issue}, t_{expire}, \Lambda_{VRT} \rangle$$

Here, t_{issue} and t_{expire} define the validity period (typically 10 minutes), and Λ_{VRT} defines the rate-limit parameters.

b) *Revocation Enforcement and Propagation*: Revocation is modeled as a state transition function $S : \mathcal{C} \rightarrow \{Active, Revoked\}$. The transition to *Revoked* occurs systematically upon the submission of a valid cryptographic proof Π from a Pocket Ledger. The Root Ledger validates Π by verifying signatures from a supermajority ($2f_{val} + 1$) of the escalating region’s validators:

$$S(C_i) \leftarrow Revoked \iff Verify(\Pi, \mathcal{V}_{val}^{(k)}) = True$$

Once committed, the revocation state is immutable. To ensure system availability during network partitions or rural operation, vehicles maintain a local Revocation Cache of the Root Ledger state. If connectivity to the Root Ledger is lost, the vehicle enforces the last known valid state.

Propagation utilizes a hybrid push-pull mechanism over Cellular V2X (C-V2X). Vehicles subscribe to lightweight WebSocket events for immediate alerts and poll for state deltas every 5 seconds to reconcile local caches. Assuming standard LTE/5G availability, this approach ensures that global revocation propagates to the network edge with sub-5-second latency, offering near real-time propagation compared to the periodic update cycles of traditional Certificate Revocation Lists (CRLs).

c) *Cross-Regional Attack Correlation*: To mitigate ‘‘region hopping’’ attacks where mobile adversaries exploit geographic isolation, the Root Ledger aggregates evidence from all Pocket Ledgers via periodic state anchoring. Let $\mathcal{E}_j$ be the set of distinct regions that have recorded a confirmed misbehavior event $D_{confirm}(v_j)$, and let $T_j^{(k)}$ denote the current trust score of vehicle v_j within Pocket Ledger L_k .

The global correlation function triggers revocation if the pseudonym is implicated in multiple regions OR if any single region demands revocation:

$$(|\mathcal{E}_j| \geq 2) \vee (\exists L_k : T_j^{(k)} < T_{revoke})$$

This mechanism ensures that an attacker moving between regions is identified and revoked globally upon the second incident, even if they fail to trigger the full trust penalty threshold in any single location. Furthermore, this correlation mathematically bounds an adversary exploiting geographic isolation to a maximum blast radius of a single unpenalized regional attack. A confirmed incident in a second distinct region immediately triggers global revocation, permanently neutralizing the colluding identities.

E. Verifiable Rate-Limit Enforcement

Secure-BSM applies a Verifiable Rate-limit Token (VRT) mechanism modeled as a cryptographic token bucket to enforce the congestion control requirements defined in [24]. This transforms rate limiting from a voluntary local policy into a cryptographically enforced network-wide constraint.

a) *Token Allocation (On-Chain State)*: The Root Ledger serves as the source of truth for rate-limit parameters. Upon registration, each pseudonym is assigned a VRT configuration:

$$C_{config} = \langle C_{max}, \lambda, t_{start} \rangle$$

where λ is the refill rate (10 Hz) and C_{max} is the burst capacity. Unlike currency systems, the live token balance is not stored on-chain to avoid transaction latency. Instead, the Ledger anchors the validation rules that neighbors must enforce.

b) *Consumption and Verification (Off-Chain Enforcement)*: Enforcement utilizes a "Shadow State" model where receivers locally track the token usage of neighbors based on the ledger's parameters. This model bridges the gap between the ≈ 1.0 s block interval and real-time attacks.

- *Sender Obligation*: The sender maintains a local token counter. It must sign every BSM with a monotonically increasing sequence number seq .
- *Receiver Verification*: Upon receiving a message from v_i , the receiver calculates v_i 's theoretical balance $B_i(t)$ using the time delta since the last accepted message (t_{last}) and the ledger-defined rate λ :

$$B_i(t) = \min(C_{max}, B_i(t_{last}) + \lambda(t - t_{last})) - 1$$

The message is accepted if and only if $B_i(t) \geq 0$. If $B_i(t) < 0$, the receiver knows v_i has exceeded the ledger-defined contract. The packet is immediately dropped to preserve local processing resources, effectively filtering DoS traffic at the network edge.

c) *Attack Mitigation Dynamics*: This mechanism ensures that a node transmitting at rate $\lambda_{attack} > \lambda$ will inevitably generate a negative balance in the view of honest receivers. The time to exhaustion is defined as:

$$t_{exhaust} = \frac{C_{max}}{\lambda_{attack} - \lambda}$$

For example, assuming a full bucket ($C_{max} = 5$) and a high-intensity flood at 50 Hz ($5 \times$ normal rate), the tokens are exhausted in 0.125 seconds. Once the shadow balance hits zero, receivers automatically drop subsequent packets, effectively filtering the DoS traffic at the network edge without requiring global consensus intervention.

TABLE II
SECURE-BSM IMPLEMENTATION CONFIGURATION PARAMETERS

Component	Value
<i>Blockchain Infrastructure (Hyperledger Besu)</i>	
Consensus Protocol	IBFT 2.0 (Proof-of-Authority)
Root Ledger Validators	4 (Infrastructure, $f = 1$)
Pocket Ledger Validators	Dynamic (Voting-based, $T_j > 90$)
Block Interval	≈ 1.0 s
Root Epoch Length	30,000 blocks
Pocket Epoch Length	1,000 blocks
<i>Network & Cryptography</i>	
BSM Broadcast Rate	10 Hz
VRT Refill Rate (λ)	10 tokens/s
VRT Bucket Capacity (C_{max})	5 tokens (Tight Burst Limit)
Signature Scheme	ECDSA secp256k1
Pseudonym Validity	600 s
<i>Simulation Environment</i>	
Mobility Engine	SUMO v1.15
Runtime Environment	Python 3.10 / Docker

TABLE III
LOCAL DETECTION ENGINE THRESHOLDS

Detector Module	Parameter	Value
Kinematic (FDIA)	Max Velocity (v_{max})	45 m/s
	Max Acceleration (a_{max})	10 m/s ²
	Max Heading Change	45°/s
Gradual FDIA	Displacement Threshold (δ_{err})	2.5 m
	Projection Model	Linear
Replay Defense	Max Clock Skew (τ_{clock})	2.0 s
	Sequence Check	Monotonic Increment
DoS Defense	Rate Limit Threshold	15 msg/s
	Detection Window (W_{dos})	1.0 s
Sybil Detection	Spatial Radius (R)	5.0 m
	Min Velocity to Trigger	2.0 m/s

IV. IMPLEMENTATION

A. Blockchain Infrastructure

1) *Root Ledger Deployment*: The Root Ledger is implemented as a persistent Hyperledger Besu network [25] operating under the IBFT2 consensus protocol. For this simulation, four infrastructure validators are deployed to tolerate one Byzantine failure ($f = 1$) though the network size is configurable. Each validator runs in a containerized environment with fixed resource allocations of 1 vCPU and 2 GB RAM, reflecting realistic automotive-edge constraints.

The network is configured with ≈ 1 s block intervals to ensure deterministic finality and low-latency state updates. Epoch length is fixed at 30,000 blocks to minimize validator churn. RPC endpoints are exposed via HTTP and WebSocket interfaces to support transaction submission and event-driven synchronization. The genesis configuration initializes prefunded accounts and enforces bounded block gas limits to stabilize throughput. The specific configuration parameters for the consensus protocol and block intervals are detailed in Table II.

A suite of smart contracts governs global security functions. VehicleRegistry manages pseudonym reg-

istration and lifecycle transitions. `VRTManager` enforces cryptographic rate limits via token allocation parameters. `RevocationManager` maintains immutable revocation records anchored by Merkle proofs. `ReputationVault` manages the trust scores and validator eligibility criteria, enforcing the probationary initialization logic ($T_{init} = 50$). Cross-ledger evidence anchoring is realized through `PocketAnchor`, while `EvidenceHub` stores hashed forensic metadata for long-term auditability.

2) *Pocket Ledger Instantiation and Validator Selection:* Pocket Ledgers are instantiated as ephemeral IBFT2 networks scoped to individual simulation scenarios. Crucially, the Pocket Ledger does not exist as a physical infrastructure node or at a static geographic location. Rather, it is deployed as an ephemeral, decentralized software overlay network that “lives” entirely within the containerized Hyperledger Besu clients (allocated 1 vCPU and 2GB RAM) running concurrently on the On-Board Units (OBUs) of the participating vehicles. The network dynamically instantiates over the V2V radio interface and dissolves as vehicles disperse. Unlike the Root Ledger, validator membership is dynamic and composed exclusively of participating vehicles, enabling the vehicle-as-validator paradigm.

Validator selection implements the dynamic voting logic described in Section III. Candidate vehicles submit participation requests and are evaluated based on the geographic radius r_{geo} and their current Trust Score T_j . Instead of financial staking, the system uses a “Reputation Stake” where only vehicles with $T_j > 90$ are eligible to propose blocks.

Validator set size adapts to vehicle density to maintain optimal Byzantine fault tolerance ($3f_{val} + 1$). In scenarios with sufficient density, the network dynamically scales the validator set size to accommodate local traffic volume. In extremely sparse environments where the minimum consensus quorum (typically $n < 4$) cannot be met, the Pocket Ledger consensus is temporarily suspended. During these intervals, the system reverts to a “Local-Only” mode where vehicles rely exclusively on their internal detection engines until sufficient peers arrive to re-establish the validator set. Pocket Ledgers use ≈ 1 s block intervals and short epochs of 1,000 blocks to allow rapid validator rotation and failure recovery. Validator admission and removal are executed through on-chain voting (requiring $2f_{val} + 1$ approvals), preserving Byzantine safety during membership changes.

3) *Collaborative Evidence Aggregation:* Misbehavior reports are aggregated by the `PocketDetectionHub` contract. Each report encodes the suspect pseudonym, attack type, evidence hash, and observation timestamp. The contract maintains per-suspect state, specifically tracking the set of unique reporting peers.

To prevent Sybil amplification, duplicate reporting is suppressed using epoch-based reporter tracking. Consensus is reached when the count of unique reporters satisfies the reporting quorum (τ_{rep}), at which point the suspect pseudonym is flagged and corresponding events are emitted. These thresholds are governance-adjustable to accommodate varying threat

environments.

B. Cryptographic Identity Management

1) *Pseudonym Generation and Storage:* Each vehicle pre-generates a pool of ECDSA secp256k1 key pairs. Pseudonym identifiers are derived as Keccak-256 hashes of public keys [23] and represented as Ethereum-compatible addresses, eliminating auxiliary identity mappings. Private keys are stored locally using authenticated encryption (AES-256-GCM). Key-store files persist on disk and are decrypted only transiently during signing operations, minimizing key exposure.

2) *Certificate Lifecycle and Rotation:* Pseudonym certificates bind public keys to validity intervals and initial rate-limit quotas. Certificates are valid for 600 s and rotated periodically to ensure unlinkability. Rotation triggers re-registration and quota allocation on the Root Ledger. Expired certificates are rejected during BSM verification, preventing replay across pseudonym epochs.

3) *BSM Authentication Pipeline:* Outgoing BSMs are serialized deterministically to ensure canonical hashing. Messages are signed using ECDSA over Keccak-256 digests, incorporating the current VRT sequence number. Receivers verify signatures via public-key recovery and address matching against claimed pseudonyms, providing cryptographic authenticity without prior key exchange.

4) *Revocation Synchronization and Trust Lifecycle:* To maintain the continuity of the collaborative consensus, the Trust Score (T_j) operates as a persistent, global attribute anchored by the Root Ledger’s `ReputationVault`. A vehicle accumulates reputation across multiple driving sessions and regions over time, rather than resetting its score upon entering a new Pocket Ledger region.

To prevent synchronous infrastructure bottlenecks when verifying these global scores, vehicles maintain an in-memory cache of the Root Ledger state. This cache is asynchronously updated through hybrid synchronization. Periodic polling retrieves the full revoked pseudonym set, while WebSocket subscriptions deliver immediate revocation events. Propagation utilizes a hybrid mechanism primarily via cellular networks (LTE/5G), utilizing opportunistic RSU offloading only where available. When a vehicle attempts to join a new Pocket Ledger validator set, the existing nodes verify its eligibility ($T_j > 90$) via constant-time local cache lookups. This completely decouples real-time Pocket Ledger block production and validator admission from infrastructure communication latency, ensuring the system remains autonomous and performant at the network edge.

C. Detection Engine Realization

The misbehavior detection engine executes locally on each vehicle and processes incoming BSMs through a sequential pipeline corresponding to the detectors defined in Section III, utilizing the thresholds listed in Table III. State tracking includes position history, rate windows, and spatial clustering indexes. When a detector triggers, a signed `MisbehaviorReport` is generated and queued for submission to the Pocket Ledger.

D. Vehicle Controller Integration

Each vehicle instantiates a controller that integrates SUMO mobility data [26], cryptographic services, detection logic, and blockchain clients. The controller operates at 10 Hz, synchronized with simulation timesteps. At each timestep, the controller acquires ground-truth mobility state, constructs and signs a BSM, locally verifies the available VRT quota against the cached ledger state, and broadcasts the message to neighboring vehicles. Received BSMs undergo signature verification, revocation checking, and detection processing. Safety-critical detections trigger immediate behavioral responses such as emergency braking, while all detections are asynchronously reported to the Pocket Ledger. Blockchain interactions are non-blocking to preserve real-time vehicle operation.

To address the stringent latency requirements of physical vehicular control, the architecture explicitly decouples immediate hazard avoidance from cryptographic consensus. When the Local Detection Engine identifies an anomaly, it triggers safety-critical behavioral responses (e.g., emergency braking or trajectory alteration) in milliseconds. This primary safety loop requires zero blockchain interaction, ensuring that the 2.11 to 3.92-second Pocket Ledger consensus latency never impedes real-time vehicle control.

Simultaneously, the evidence of the detection is asynchronously reported to the Pocket Ledger. The multi-second consensus latency is strictly reserved for generating the irreversible cryptographic penalty and localized network isolation. By operating these processes on parallel tracks, Secure-BSM reduces the revocation window from minutes (as seen in traditional SCMS architectures) down to a few seconds without encumbering the millisecond-scale physical control loops. Blockchain interactions remain strictly non-blocking to preserve this real-time vehicle operation.

E. Verifiable Rate-Limit Enforcement

Rate limiting is enforced using Verifiable Rate-limit Tokens (VRTs). While the token configuration is anchored in the Root Ledger, the enforcement logic utilizes the *Optimistic Shadow State* model to ensure low latency. Each pseudonym receives a fixed token allocation corresponding to nominal BSM transmission rates (10 Hz).

Instead of relying on high-latency on-chain state updates, enforcement bridges the block interval gap through local tracking. Senders locally decrement their authorized token balance and sign the message using the authorized sequence number. Receivers validate message legitimacy by maintaining a transient shadow balance B_{local} derived from the ledger parameters. Quota exhaustion results in automatic message rejection, providing network-wide DoS mitigation independent of local detection heuristics.

F. Attack Injection Framework

Attack behaviors are implemented through a configuration-driven injection framework. Scenarios specify attack type, target vehicle, activation interval, and attack parameters. Attacks manipulate BSM contents or transmission behavior prior

TABLE IV
ATTACK INJECTION PARAMETERS AND SCENARIO DEFINITIONS

Attack Vector	Injection Logic (Deviation)	Victim Perception
Constant FDIA	$P_{tx} = P_{real} + 50m$ (Offset)	Vehicle jumps ahead
Gradual FDIA	$P_{tx}(t) = P_{real}(t) + (0.5 \text{ m/s} \times t)$	Subtle lane drift
Sybil Attack	Broadcast 5 IDs from single P_{real}	Ghost congestion
Spoofing	Broadcast using ID_{victim}	Identity conflict
DoS Flooding	Rate $f_{tx} = 50 \text{ Hz}$ (Limit: 10Hz)	Channel saturation
Replay Attack	$BSM(t) = BSM_{recorded}(t - \delta_{time})$	Teleportation backward

to signing, ensuring cryptographic validity while inducing semantic misbehavior. Implemented attacks include constant and gradual FDIA, replay, DoS flooding, Sybil identity amplification, and credential spoofing. The framework supports deterministic replay of scenarios for controlled experimentation.

V. PERFORMANCE EVALUATION

We evaluate the performance of Secure-BSM focusing on three critical dimensions: (i) detection accuracy and the reduction of false positives through collaborative consensus, (ii) system scalability under varying validator set sizes, and (iii) the effectiveness of the Verifiable Rate-limit Token (VRT) mechanism against DoS attacks.

A. Experimental Setup

The simulation environment was built using SUMO v1.15 for traffic mobility and Hyperledger Besu (v24.3.0) for the blockchain network, utilizing the IBFT 2.0 consensus mechanism. To mimic the strict resource constraints of standard On-Board Units (OBUs), each blockchain node was deployed in a Docker container restricted to 1 vCPU and 2GB RAM. This configuration demonstrates the feasibility of hosting the Pocket Ledger on commodity automotive-grade hardware without requiring specialized High-Performance Computing (HPC) clusters. The network settings simulated an IEEE 802.11p (DSRC) environment [27] with a broadcast rate of 10 Hz and a communication range of 300 meters.

B. Dataset Generation and Attack Injection

We generated a comprehensive dataset containing 79,636 BSM records using a “Twin-Run” methodology. First, baseline simulations generated ground-truth Floating Car Data (FCD). Second, a Python middleware was used to inject malicious behavior into 10% to 30% of the vehicles. Table IV details the perturbation logic applied to generate the six distinct attack vectors, modeled after standard misbehavior profiles [5].

C. Detection Performance Analysis

To quantify the security benefits of the proposed architecture, we establish the standalone *Local Engine* (detailed in Section III-B) as the baseline. We compared its raw detection metrics against the *Secure-BSM Collaborative* framework to

isolate the impact of the consensus layer. The results, summarized in Table V, validate the hypothesis that local sensors are prone to environmental noise, which consensus effectively filters.

Noise Reduction: The Local Engine baseline exhibits a notable False Positive Rate (FPR) for *Gradual FDIA* (18.4%) and *Sybil Attacks* (12.5%). This is attributed to the inherent difficulty of distinguishing slow sensor drifts or signal reflections from malicious behavior using single-point observations. Secure-BSM suppresses these errors (reducing Gradual FDIA FPR to 4.2%) by requiring a reporting quorum (τ_{rep}). This confirms that correlated agreement is a superior metric for revocation than isolated detection.

Precision-Recall Trade-off: The impact of consensus introduces a deliberate trade-off between sensitivity and safety. For subtle vectors like *Gradual FDIA*, the strict quorum requirement causes a marginal drop in Recall (0.820 $\rightarrow$ 0.805) compared to the baseline. This occurs because edge-case attacks detected by only a minority of neighbors fail to trigger the consensus threshold. However, this conservative approach drives a significant improvement in Precision (0.765 $\rightarrow$ 0.945), ensuring that the system prioritizes the protection of honest vehicles over the aggressive removal of ambiguous threats. For high-confidence attacks like Spoofing, consensus maintains near-parity with local detection (0.940 $\rightarrow$ 0.932), with the slight reduction attributed to packet loss during the voting phase, while virtually eliminating false alarms.

Temporal Bounds in High-Mobility Environments: A critical operational constraint is the ability to detect and finalize consensus on an attacker before they exit the geographic region ($r_{geo} = 150$ m). For highway driving at 80 mph (36 m/s), a vehicle traverses the 300 m diameter in ≈ 8.33 seconds. Under the *Gradual FDIA* scenario (0.5 m/s drift), the attack crosses the 2.5 m displacement threshold (δ_{err}) at exactly $t = 5.0$ seconds. Factoring in the optimal consensus latency of 2.11 seconds ($N = 10$), the collaborative confirmation completes in 7.11 seconds. This mathematical bound ensures the attacker is permanently penalized and escalated to the Root Ledger before exiting the tight operational window.

Resilience to Network Realities: While lower-layer MAC/PHY conditions such as packet loss, hidden terminals, and RF channel fading inevitably affect BSM delivery, Secure-BSM is architecturally resilient to these network realities through built-in redundancy. Because BSMs are broadcast at 10 Hz, an attacker transmits dozens of malicious messages within a short window. Even under severe channel degradation (e.g., 50% packet loss), the probability that at least $\tau_{rep} \geq 3$ independent neighbors successfully receive and report the anomaly remains exceptionally high, ensuring liveness is preserved. Furthermore, filtering out the environmental anomalies caused by these physical realities is the primary function of the Pocket Ledger, driving the FPR down to 4.2% by effectively decorrelating perspective-specific hardware errors.

TABLE V
DETECTION ACCURACY: LOCAL ENGINE VS. COLLABORATIVE CONSENSUS

Attack Vector	Precision		Recall		FPR	
	Local	Consensus	Local	Consensus	Local	Consensus
Replay Attack	0.945	0.992	0.980	0.975	5.2%	0.8%
Constant FDIA	0.912	0.988	0.965	0.960	8.5%	1.2%
Gradual FDIA	0.765	0.945	0.820	0.805	18.4%	4.2%
Spoofing	0.890	0.985	0.940	0.932	7.1%	1.5%
Sybil Attack	0.840	0.978	0.910	0.905	12.5%	2.1%

TABLE VI
CONSENSUS LATENCY AND THROUGHPUT VS. VALIDATOR SET SIZE

Validator Set (N)	Latency (s)	Throughput (TPS)	Std. Dev.
4	1.07	439	0.08
7	1.52	385	0.17
10	2.11	305	0.10
20	3.92	218	0.18

D. System-Level Evaluation and Overhead

We measured the Time-to-Confirmation and throughput across varying validator set sizes ($N_{val} \in [4, 20]$). The throughput benchmark utilized synthetic MisbehaviorReport transactions (payload size ≈ 200 bytes, execution cost $\approx 45,000$ gas) submitted to a Pocket Ledger instance. The block gas limit was set to 80×10^6 to isolate consensus latency from block capacity constraints. As detailed in Table VI, the system exhibits linear latency growth consistent with IBFT 2.0 complexity. The results identify a clear operational boundary at $N_{val} = 10$. Up to 10 validators, the system maintains a consensus latency of ≤ 2.11 s, well within the 3-second safety window for revocation [28]. Beyond this point (e.g., at 20 validators), latency degrades to nearly 4 seconds, suggesting that $N \approx 10$ is the optimal cluster size for the Pocket Ledger.

Communication Overhead Analysis: For a validator set of size N , the protocol generates approximately $2N(N-1)$ administrative messages per block. At the optimal operational point of $N = 10$ (derived from our latency benchmarks), this results in ≈ 180 messages per second (given a 1.0 s block interval). Assuming a conservative administrative message size of 200 bytes (containing BFT headers and ECDSA signatures), the total consensus overhead is $180 \text{ msgs/s} \times 200 \text{ bytes} \times 8 \text{ bits} \approx 288 \text{ kbps}$. Given the standard IEEE 802.11p (DSRC) channel capacity of 6 Mbps (and significantly higher for C-V2X), the consensus traffic consumes approximately 4.8% of the available bandwidth. This confirms that the Pocket Ledger operates efficiently without congesting the safety-critical BSM channel.

E. DoS Mitigation Effectiveness

We evaluated the Verifiable Rate-limit Token (VRT) mechanism under a 50 Hz flooding attack ($5 \times$ the legitimate rate). Table VII presents the ablation study comparing system performance with and without VRT enforcement. In the absence of VRT, the communication channel saturates, reducing legitimate message throughput to 57.9%. When VRT is enabled, the token bucket mechanism (10 Hz refill, 5-token burst limit)

TABLE VII
DoS MITIGATION: ABLATION STUDY (50 HZ ATTACK)

Metric	Without VRT	With VRT	Impact
Legitimate Throughput	57.9%	88.9%	+31.0 pp
Attack Containment	Unlimited	<0.2s	Strictly Bounded

successfully throttles the attacker. While strict rate-limiting introduces a minor processing overhead, it restores legitimate throughput to 88.9%, preserving network availability for safety-critical messages.

VI. DISCUSSION

A. Architectural Superiority and Impact

The primary advantage of Secure-BSM lies in its ability to decouple misbehavior detection from fixed infrastructure. Unlike traditional revocation systems that suffer from minute-level latencies due to backend processing, our *Vehicle-as-Validator* approach enables millisecond-level decision-making at the network edge. This architectural independence delivers two critical performance gains.

First, the system achieves significantly enhanced detection precision. By aggregating observations from multiple peers via the Pocket Ledger, Secure-BSM filters out the stochastic sensor noise that plagues standalone detectors. As evidenced by our evaluation, this collaborative verification reduces false positive rates by 77% to 85% across tested attack vectors. This drastic reduction is not coincidental but structural: local false positives typically stem from perspective-specific errors (e.g., GPS drift, temporary occlusion) that affect single observers. By enforcing a reporting quorum ($\tau_{rep} \geq 3$), the consensus mechanism effectively decorrelates these independent errors, ensuring that transient sensor glitches do not propagate into confirmed revocations. This capability is particularly valuable in handling high-noise scenarios (e.g., Gradual FDIA), preventing the wrongful revocation of honest vehicles—a crucial requirement for liability-sensitive autonomous driving.

Second, the introduction of Verifiable Rate-limit Tokens (VRT) fundamentally transforms DoS defense. The VRTs enforce bandwidth limits using a cryptographically anchored shadow state. This ensures that attackers are contained in < 0.2 seconds, guaranteeing channel availability for legitimate safety-critical messages even under heavy adversarial load.

B. Operational Boundaries and Limitations

While the architecture offers robust security guarantees, its effectiveness is bound by specific operational conditions. A primary constraint is the dependency on a critical mass of participants. The consensus protocol requires a minimum validator set ($n \geq 3f_{val} + 1$) to maintain Byzantine fault tolerance; in sparse traffic scenarios, such as rural highways at night, the system lacks sufficient peers to form a Pocket Ledger. In these instances, the protocol gracefully degrades to *Local-Only* detection, temporarily trading collaborative assurance for standalone availability.

This behavior is a deliberate feature of graceful degradation, not a system vulnerability. The necessity of consensus scales dynamically with traffic density. In dense environments, sensor noise, occlusion, and the risk of cascading traffic disruptions from false-positive emergency braking are at their highest, making collaborative verification strictly essential. Conversely, in sparse rural environments, encounters are largely isolated, sensor line-of-sight is unobstructed, and the risk of cascading platoon failure is minimal. In these conditions, the standalone Local Engine—which enforces rigorous kinematic, temporal, and cryptographic constraints—operates with significantly less environmental noise and is sufficient to maintain baseline vehicular safety until the vehicle merges into denser traffic where collaborative verification becomes strictly necessary.

To prevent Sybil attacks, the system enforces a strict entry barrier: only nodes with a high trust score ($T_j > 90$) can become validators. This successfully blocks immediate attacks from newly created identities. We acknowledge, however, that a determined adversary could theoretically circumvent this by “farming” reputation over time—behaving honestly for long periods just to gain access. We assume that the significant time and effort required to groom these identities (e.g., 16 hours for a minimal cluster takeover) make such an attack economically and practically irrational for standard disruption scenarios.

C. Future Research Directions

To further harden the architecture against these edge cases, future work will focus on:

- 1) *Privacy-Preserving Reporting*: Integrating Zero-Knowledge Proofs (ZK-SNARKs) [29] to allow vehicles to submit misbehavior reports without revealing their pseudonym history, eliminating metadata leakage risks.
- 2) *Reputation-Weighted Voting*: Transitioning from a “one-node-one-vote” model to a weighted voting scheme where voting power is proportional to reputation age, rendering Sybil farming economically impractical.
- 3) *Adaptive Geographic Partitioning*: Developing mechanisms to dynamically subdivide Pocket Ledgers in highly dense scenarios. This ensures that validator sets remain small and responsive even when hundreds of vehicles occupy the same physical intersection.

VII. CONCLUSION

This paper introduced **Secure-BSM**, a collaborative misbehavior detection framework utilizing dynamic Pocket Ledgers to eliminate reliance on pervasive roadside infrastructure. Our SUMO and Hyperledger Besu evaluation demonstrates that Secure-BSM improves detection precision for Gradual FDIA and Sybil attacks by 18.0% and 13.8%, respectively, while reducing false positive rates by 77%–85% across all vectors. Furthermore, the Verifiable Rate-limit Token (VRT) mechanism successfully contains DoS flooding in under 0.2 seconds. Ultimately, transferring consensus logic to the network edge is both highly feasible on commodity automotive hardware and essential for the next generation of resilient intelligent transportation systems.

REFERENCES

- [1] "Ieee standard for wireless access in vehicular environments—security services for application and management messages," *IEEE Std 1609.2-2022 (Revision of IEEE Std 1609.2-2016)*, pp. 1–349, 2023.
- [2] S. M. Mostaq Hossain, S. Banik, T. Banik, and A. M. Shibli, "Survey on security attacks in connected and autonomous vehicular systems," in *2023 IEEE International Conference on Computing (ICOCO)*, 2023, pp. 295–300.
- [3] B. Brecht, D. Theriault, A. Weimerskirch, W. Whyte, V. Kumar, T. Hehn, and R. Goudy, "A security credential management system for v2x communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 12, pp. 3850–3871, 2018.
- [4] G. Scopelliti, C. Baumann, F. Alder, E. Truyen, and J. T. Mühlberg, "Efficient and timely revocation of v2x credentials," in *Proceedings of the 2024 Network and Distributed System Security Symposium (NDSS'24)*. Internet Society, 2024.
- [5] S. So, P. Sharma, and J. Petit, "Integrating plausibility checks and machine learning for misbehavior detection in vanet," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018, pp. 564–571.
- [6] J. Kamel, A. Kaiser, I. ben Jemaa, P. Cincilla, and P. Urien, "Catch: A confidence range tolerant misbehavior detection approach," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, 2019, pp. 1–8.
- [7] P. Sharma and H. Liu, "A machine-learning-based data-centric misbehavior detection model for internet of vehicles," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4991–4999, 2021.
- [8] F. Dutra, K. Bonfim, C. Siqueira, L. A. Pereira, A. Santos, and R. I. Meneguette, "Dismiss-bsm: an architecture for detecting position spoofing in basic safety messages," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022, pp. 3381–3386.
- [9] A. Boualouache and T. Engel, "A survey on machine learning-based misbehavior detection systems for 5g and beyond vehicular networks," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1128–1172, 2023.
- [10] A. Boualouache, R. Soua, and T. Engel, "Sdn-based misbehavior detection system for vehicular networks," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5.
- [11] W. Ahmed, W. Di, and D. Mukathe, "Privacy-preserving blockchain-based authentication and trust management in vanets," *IET Networks*, vol. 11, no. 3-4, pp. 89–111, 2022.
- [12] J. Zhang, W. Lin, X. Shang, Y. Zeng, and Y. Yang, "Secured data sharing and storage system for intelligent transportation system via lightweight blockchain," in *2025 IEEE 50th Conference on Local Computer Networks (LCN)*, 2025, pp. 1–8.
- [13] A. Tesei, D. Lattuca, M. Luise, P. Pagano, J. Ferreira, and P. C. Bartolomeu, "A transparent distributed ledger-based certificate revocation scheme for vanets," *Journal of Network and Computer Applications*, vol. 212, p. 103569, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1084804522002107>
- [14] R. Boopathi, P. Ramakrishnan, M. Jamuna Rani, P. Vijayakumar, S. Siva Kumar, and T. Sathya, "A blockchain-based multi-metric trust management framework for secure and scalable vehicular ad-hoc networks," in *2025 Third International Conference on Augmented Intelligence and Sustainable Systems (ICAIS)*, 2025, pp. 1603–1609.
- [15] V.-L. Nguyen, P.-C. Lin, and R.-H. Hwang, "Enhancing misbehavior detection in 5g vehicle-to-vehicle communications," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9417–9430, 2020.
- [16] S. Gyawali and Y. Qian, "Misbehavior detection using machine learning in vehicular communication networks," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.
- [17] H.-Y. Hsu, N.-H. Cheng, and C.-W. Tsai, "A deep learning-based integrated algorithm for misbehavior detection system in vanets," in *Proceedings of the 2021 ACM International Conference on Intelligent Computing and Its Emerging Applications*, ser. ACM ICEA '21. New York, NY, USA: Association for Computing Machinery, 2022, p. 53–58. [Online]. Available: <https://doi.org/10.1145/3491396.3506509>
- [18] A. Lochbihler, A. Husseinat, M. Ma, M. Abuibaid, J. S. Huang, and B. Zhu, "Misbehaviour detection of 5g-connected vehicles using deep learning," in *2023 IEEE 20th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*, 2023, pp. 646–651.
- [19] K. A. Bonfim, F. D. S. Dutra, C. E. T. Siqueira, R. I. Meneguette, A. L. Dos Santos, and L. A. P. Júnior, "Federated learning-based architecture for detecting position spoofing in basic safety messages," in *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, 2023, pp. 1–5.
- [20] A. Upreti, D. B. Rawat, and J. Li, "Privacy preserving misbehavior detection in iov using federated machine learning," in *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, 2021, pp. 1–6.
- [21] Z. Abou El Houda, H. Moudoud, B. Brik, and L. Khokhi, "Blockchain-enabled federated learning for enhanced collaborative intrusion detection in vehicular edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 7, pp. 7661–7672, 2024.
- [22] H. Moniz, "The istanbul bft consensus algorithm," *arXiv preprint arXiv:2002.03613*, 2020.
- [23] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.
- [24] D. T. Committee *et al.*, *On-board system requirements for V2V safety communications*. SAE International, 2020.
- [25] Hyperledger Foundation, *Hyperledger Besu Documentation*, 2023, accessed: 2026-01-01. [Online]. Available: <https://besu.hyperledger.org/>
- [26] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wiessner, "Microscopic traffic simulation using sumo," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2575–2582.
- [27] D. Jiang and L. Delgrossi, "Ieee 802.11 p: Towards an international standard for wireless access in vehicular environments," in *VTC Spring 2008-IEEE vehicular technology conference*. IEEE, 2008, pp. 2036–2040.
- [28] J. B. Kenney, "Dedicated short-range communications (dsrc) standards in the united states," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.
- [29] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *2014 IEEE symposium on security and privacy*. IEEE, 2014, pp. 459–474.