V-PASS: Sybil-Resistant Pseudonym Self-Provisioning for V2X

Hexuan Yu
Virginia Tech
Arlington, VA, USA
hexuanyu@vt.edu

Md Mohaimin Barat
Virginia Tech
Arlington, VA, USA
barat@vt.edu

Shaoyu Li
Virginia Tech
Arlington, VA, USA
shaoyuli@vt.edu

Md Hasan Shahriar
Virginia Tech
Arlington, VA, USA
hshahriar@vt.edu

Yang Xiao
University of Kentucky
Lexington, KY, USA
xiaoy@uky.edu

Panagiotis Papadimitratos
KTH Royal Institute of Technology
Stockholm, Sweden
papadim@kth.se

Y. Thomas Hou
Virginia Tech
Blacksburg, VA, USA
hou@vt.edu

Wenjing Lou
Virginia Tech
Arlington, VA, USA
wjlou@vt.edu

Abstract

Standardized Vehicle-to-everything (V2X) security architectures, such as the Secure Credential Management System (SCMS) in the U.S. and the C-ITS Security Credential Management System (CCMS) in Europe, protect driver privacy through large-scale batch issuance of short-lived pseudonym certificates. At national deployment scale, minute-level rotation across hundreds of millions of vehicles results in trillion-scale annual certificate management, creating substantial backend complexity, storage burden, and persistent cellular connectivity requirements.

This paper presents V-PASS, a hardware-rooted architecture that enables vehicles to derive unlinkable pseudonyms locally from a single enrollment credential, eliminating periodic batch provisioning from the Vehicular PKI (VPKI). A Secure Element (SE) enforces a window-level authorization policy and produces a hardware-bound attestation, ensuring that at most one valid pseudonym can be derived per window even under ECU software compromise, thereby providing *ex-ante* resistance against Sybil attacks.

V-PASS adopts a split-trust design that confines lightweight authorization to the SE while delegating computationally expensive cryptographic proofs to the ECU. The resulting window object enables stateless verification for receivers while preserving cross-window unlinkability and conditional accountability.

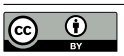
We implement V-PASS on a Raspberry Pi 5 emulating constrained automotive hardware. Evaluation shows a verification latency of **0.30 ms** per message, supporting real-time V2X safety messaging, while achieving a **1750**× reduction in pseudonym storage compared to traditional VPKIs, demonstrating practical feasibility for large-scale deployment.

CCS Concepts

• **Security and privacy**;

Keywords

V2X Security; Sybil Resistance; Privacy; Authentication



WiSec '26, Saarbrücken, Germany

ACM ISBN 979-8-4007-2201-1/2026/06
https://doi.org/10.1145/3765613.3811692

ACM Reference Format:
Hexuan Yu, Md Mohaimin Barat, Shaoyu Li, Md Hasan Shahriar, Yang Xiao, Panagiotis Papadimitratos, Y. Thomas Hou, and Wenjing Lou. 2026. V-PASS: Sybil-Resistant Pseudonym Self-Provisioning for V2X. In *Proceedings of the 19th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '26), June 30-July 03, 2026, Saarbrücken, Germany.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3765613.3811692

1 Introduction

Vehicle-to-Everything (V2X) communication forms the safety-critical data plane for future Cooperative Intelligent Transportation Systems (C-ITS). Vehicles broadcast high-frequency authenticated state beacons, such as Basic Safety Messages (BSMs) or Cooperative Awareness Messages (CAMs), to facilitate advanced driver assistance systems (ADAS), including real-time collision avoidance, intersection movement assistance, and automated traffic coordination, which can extend to enabling cooperative perception in autonomous driving [2, 3, 18]. As these messages contain precise kinematics (location, speed, heading), signing them with a single, static certificate would allow any passive eavesdropper to easily track a vehicle's movements over time. To prevent mass tracking of motorists, these communications must simultaneously ensure cryptographic *unlinkability*, preventing passive observers from correlating messages to a specific vehicle's long-term identity [10].

1.1 Problems

Standard Vehicular PKI (VPKI) architectures, such as the U.S. SCMS [8, 35] and the European CCMS [16, 17], achieve privacy through the issuance of massive batches of short-lived pseudonym certificates (PCs) [5], which are frequently rotated to thwart the spatial-temporal tracking of individual vehicles. Under this infrastructure-driven model, vehicles must periodically fetch and store hundreds or thousands of certificates from the backend Pseudonym Certification Authorities (PCAs) and rotate among them during the operation. This model introduces a structural scalability–privacy trade-off. To preserve unlinkability in dense urban traffic, pseudonyms must rotate frequently [25, 29]. Frequent rotation directly increases the number of certificates issued, provisioned, indexed, and eventually revoked per vehicle. However, provisioning larger batches amplifies backend workload, accelerates certificate revocation list (CRL) growth [37, 38], and increases database maintenance costs at the

national scale. On the vehicle side, storing and indexing large pseudonym pools imposes non-trivial storage and lookup overhead on resource-constrained On-Board Units (OBUs). Conversely, limiting the number of pre-fetched certificates reduces infrastructure burden but weakens privacy guarantees when rotation frequency becomes insufficient for high-density deployments [24].

At a nationwide scale, even conservative assumptions illustrate the magnitude of this challenge: assuming an average daily commute across a fleet of 350 million vehicles [18] and minute-level pseudonym lifetimes [4], a VPKI must manage approximately 1.5×10^{12} pseudonyms annually, a volume that exceeds traditional web PKIs [33] by several orders of magnitude. Vehicles must maintain stable, high-bandwidth cellular connectivity to replenish certificate pools, which are often unreliable in rural areas or congested environments. These operational and economic tensions have motivated recent research into self-generated pseudonym protocols [9, 19] in V2X. These designs rely on an Enrollment Authority (EA) to issue a long-term Enrollment Credential (EC), while a Tracing Authority (TA) supports conditional privacy recovery. Vehicles then derive pseudonyms locally, eliminating periodic provisioning and reducing infrastructure dependence.

However, local pseudonym derivation alone does not resolve the critical problem of **Sybil control**. If a vehicle's Electronic Control Unit (ECU) is compromised—an increasingly realistic threat given the growing connectivity of automotive infotainment, telematics, and over-the-air update stacks [14, 27]—an adversary may exploit enrollment material to generate multiple valid pseudonyms concurrently. This enables classical Sybil attacks [13], where a single physical vehicle presents itself as multiple independent ghost vehicles in V2X space.

Prior work attempts to mitigate this risk primarily through *post-facto* detection mechanisms, such as trajectory consistency checks or machine-learning-based analysis of vehicular communication patterns [31, 32], or through backend correlation of authentication transcripts [9, 19]. However, V2X authentication is inherently decentralized: roadside units (RSUs) and peer vehicles make localized, real-time decisions without global visibility. In safety-critical contexts such as Cooperative Adaptive Cruise Control (CACC) or high-speed platooning [36], vehicles make millisecond-level control decisions based on peer telemetry. Multiple authenticated identities can therefore distort traffic perception, simulate phantom congestion, and trigger unsafe reactions—even though each identity appears cryptographically valid.

1.2 Our Approach

To eliminate large-scale pseudonym provisioning while preventing uncontrolled Sybil identity generation, we present V-PASS, a hardware-assisted pseudonym self-provisioning architecture that guarantees **one valid pseudonym per vehicle per window** (e.g., 2 to 5 minutes in practical V2X deployments), even under ECU software compromise.

V-PASS adopts a **split-trust architecture** that separates the rich software environment of the application processor (ECU) from a minimal trusted hardware anchor. Modern vehicular platforms already integrate hardware roots of trust, including Trusted Platform Module (TPM 2.0)-class devices, Secure Elements (SEs), and automotive HSMs for lifecycle key protection and V2X authentication [20, 30, 34]. Rather than treating this hardware merely as a passive key vault, V-PASS co-designs the cryptographic protocol around its constraints.

Lightweight pairing-free operations are executed inside the isolated SE, while computationally expensive cryptographic proofs are constructed on the powerful but untrusted ECU. A hardware-generated SE attestation is cryptographically bound to the ECU-side proof through an algebraic split construction, producing a compact authorization transcript. The resulting **Window Object** guarantees that at most one valid pseudonym can be derived per window—even if the ECU is fully compromised—thereby providing ***ex-ante* Sybil prevention**. At the same time, the construction supports stateless verification, deterministic revocation tagging, and conditional accountability for malicious vehicles while preserving cross-window unlinkability for honest users.

Contributions. Our primary contributions are:

- **Hardware-Enforced Sybil Prevention:** We design a split-trust architecture that decouples policy enforcement from proof generation. This ensures that even under full software compromise, an ECU remains strictly bounded by the rate-limiting policies enforced within the vehicle's SE.
- **Algebraically-Aligned Protocol Design:** We introduce a multi-domain cryptographic construction that enables pairing-free SEs to interoperate with pairing-heavy ECU proofs. Our design utilizes an algebraic split that binds SE-based attestations to pairing-friendly group elements, achieving high security without exceeding embedded hardware constraints.
- **Efficient Revocation and Stateless Verification:** We integrate deterministic per-window revocation tags that allow verifiers to identify blacklisted vehicles in a single pass. This mechanism preserves cross-window unlinkability for honest users while providing the "stateless" verification required for high-speed V2X safety applications.
- **System Implementation and Performance Evaluation:** We implement V-PASS on an embedded platform (Raspberry Pi 5) representative of modern automotive hardware. Evaluation results show a steady-state verification latency of **0.30 ms** and a **1750**× storage reduction compared to batch-based methods, confirming its practical feasibility.

The remainder of the paper is organized as follows. Section 2 provides background and related work. Section 3 presents the system and threat model, and Section 4 explains the design. Section 5 reviews preliminaries, and Section 6 specifies the protocol. Finally, Sections 7 and 8 provide the analysis and evaluation.

2 Related Work

Standard V2X deployments rely on a VPKI in which dedicated Pseudonym CAs periodically issue batches of short-lived certificates. Extensive prior work studies the architecture, deployment, and scalability of such infrastructures, including pseudonym provisioning, credential distribution, and certificate revocation management [5, 11, 25, 26]. These systems allow vehicles to rotate

among pre-issued credentials to achieve unlinkability, but introduce significant operational complexity, including periodic certificate replenishment, large-scale certificate revocation list distribution, and privacy guarantees that depend on the number of available pseudonyms. Several works, therefore, explore optimizations of VPKI credential management and deployment architectures [22, 28, 39]. A related line of work considers hybrid approaches that combine infrastructure-issued credentials with locally generated pseudonyms, enabling vehicles to operate when connectivity to the VPKI is unavailable [23].

A small body of recent work explores local pseudonym derivation to eliminate the dedicated pseudonym issuance authority. PSP [9] derives revocable pseudonyms from anonymous credentials with ring-based linkability tags, while Vehicular Direct Anonymous Attestation (DAA) [19] uses basename-bound DAA proofs to generate unlinkable ECDSA pseudonyms. In both designs, Sybil resistance is achieved through linkability-based *post-facto detection*: vehicles may generate multiple pseudonyms, and misuse is identified only if a verifier (or coordinating authority) collects and compares multiple transcripts. Such detection-oriented designs face inherent operational limitations in high-speed, distributed V2X settings, where no single verifier has global visibility and real-time safety decisions cannot depend on backend aggregation. In contrast, V-PASS enforces the issuance invariant at generation time, ensuring that multiple valid identities per window cannot arise in the first place, leading to sybil-resistant pseudonym self-provisioning architecture.

3 System and Threat Model

3.1 Network Entities

The V-PASS ecosystem consists of the following entities:
EA: A trusted party that registers vehicles and issues long-term ECs upon verifying their identity.
TA: An offline entity holding the master decryption key to recover identities from traceability ciphertexts in case of misbehavior.
ECU: The vehicle's main computational platform. It executes complex pairing-based cryptography and NIZK proofs but has a large attack surface.
SE: A hardware trust anchor (e.g., automotive HSM or TPM) providing protected key storage and atomic pairing-free primitives. Such components are widely deployed in modern vehicle platforms to protect cryptographic keys and support secure boot or firmware integrity mechanisms [20, 21, 30]. They typically expose monotonic counters or trusted hardware clocks backed by non-volatile state for replay and rollback protection. In V-PASS, the SE generates hardware-bound attestations that constrain pseudonym derivation at the window level.
Verifiers (RSUs/Peers): Resource-constrained nodes that perform real-time, stateless verification of broadcast packets.

3.2 Threat Model

Adversary Model: We consider a powerful Dolev-Yao adversary that controls the network. The adversary may achieve full software compromise of the vehicle's **ECU**, enabling it to extract any enrollment secrets (including m_{id}) and to manipulate any ECU-side logic. However, the adversary **cannot** violate the hardware isolation of the **SE**: it cannot extract the master secret s, forge attestations, or rollback monotonic counters. The **TA** and **EA** are assumed to be honest and non-colluding.

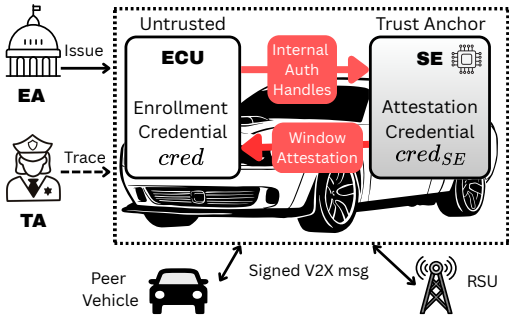


Figure 1: The V-PASS split-trust architecture. Vehicles derive pseudonyms locally from an Enrollment Credential (EC) issued by the EA. A SE enforces window-level authorization and produces a hardware attestation, while the ECU constructs the cryptographic window object WObj_w. During the active window, high-rate V2X messages are signed using the SE-resident pseudonym key and verified statelessly by peer vehicles or RSUs.

3.3 Security and Privacy Goals

(1) **G1 — Authorization Correctness:** Legitimately registered vehicles produce (via EC) accepted authentication objects.
(2) **G2 — Hardware Rate Limiting:** A vehicle with one genuine SE obtains at most one authorization per window (ctx, w), even under ECU compromise.
(3) **G3 — Anonymity and Unlinkability:** For honest vehicles, transcripts from distinct windows $w \neq w'$ remain anonymous and computationally unlinkable to any party except for the TA.
(4) **G4 — Conditional Accountability:** Upon misbehavior, the TA can recover the enrollment identity Y_{id} from an accepted transcript.

4 System Design and Overview

The core architectural choice in V-PASS is a **split-trust framework** (Figure 1) that separates *policy enforcement* from *proof construction*. This section details the execution flow and the design principles motivating our construction.

4.1 Execution Overview

V-PASS operates across two decoupled asynchronous loops.

Window authorization (low-rate loop). At the start of a new temporal window w (e.g., every 5 minutes) for a specific operational service context ctx (e.g., BSM broadcast), the ECU must request authorization from the SE. The SE autonomously derives the current window w using a trusted hardware time source or a monotonic counter maintained inside the SE, checking this against its persistent internal state $\mathsf{lastW}[\mathsf{ctx}]$. This crucial check prevents replay attacks or rapid-fire requests. If the policy validates, it updates the state and outputs a lightweight, pairing-free signature σ_{SE} on the tuple $(w, \mathsf{ctx}, \mathsf{nym}_w)$. Operating only once per window (e.g., every

five minutes), this low-rate loop completely shields the SE's severely constrained processor from high-frequency broadcast demands.

Window object construction (ECU-side). Upon receiving the hardware authorization, the powerful but untrusted ECU assumes the heavy cryptographic lifting. Using its long-term enrollment credential cred and the SE's attestation output, the ECU mathematically constructs a reusable **Window Object** (WObj_w). As formalized in Section 6, this process securely binds the hardware attestation, the ElGamal traceability ciphertexts intended for the TA, and a Fiat–Shamir NIZK proof π into a single, compact, self-contained artifact. The ECU executes these pairing-based zero-knowledge proofs entirely in fast application software.

Payload authentication (high-rate loop). Once authorized, the ECU enters the active broadcast phase. For each actual payload generated during the window w (e.g., at 10 Hz for typical V2X safety beacons), the ECU briefly invokes the SE to obtain a standard ECDSA signature σ_P over the message hash using the window-specific pseudonym key. The ECU then attaches the pre-computed WObj_w to outbound packets periodically. Because the entire proof of authorization is physically bound within WObj_w, peer receivers can validate incoming packets statelessly, requiring no matching against active sessions or persistent memory overhead per sender.

Freshness and window parameters. The window index w serves as a coarse-grained freshness label. We assume that verifiers accept packets only for windows consistent with their local time. The window duration Δ is a tunable parameter: it bounds the within-window linkability and affects amortization (Section 8).

4.2 Design Rationale and Principles

Bridging the asymmetric hardware divide. This split is motivated by automotive reality. Practical V2X platforms include an HSM/SE for key protection [16, 20, 30, 34], yet these units are typically constrained to pairing-free operations. Conversely, pairing-heavy ZK operations are better handled by the main ECU. V-PASS keeps the SE computation minimal (gating and ECDSA) and offloads proving to the ECU.

Ex-Ante Sybil Defense. Unlike software-only self-provisioning schemes that rely on reactive backend correlation to detect Sybil attacks, V-PASS moves control to preventive *source-side enforcement*. By decoupling security guarantees from network latency and backend infrastructure availability, V-PASS ensures immediate protection. Under the threat model, even a compromised ECU must present a valid SE attestation; because the SE enforces a strict one-authorization-per-context policy internally, Sybil issuance is blocked at the physical chokepoint before a single rogue packet is ever broadcast into the local RF medium. This fundamentally neuters the ability of malware to spam the network with localized phantom identities.

Algebraic alignment. To bind these domains without cross-curve proof machinery, our protocol uses algebraic alignment. The ECU proves identity statements in the pairing-friendly domain and includes the SE output in the same Fiat–Shamir transcript. This yields a single consistent object for verifiers, keeping receiver-side verification stateless.

Clean failure model. The trust boundary between SE and ECU ensures that compromise of ECU software does not expose the SE's policy state. Modern automotive SEs and HSMs are typically fortified with substantial anti-tamper mechanisms, holding certifications such as Common Criteria EAL5+ [15] to resist power-analysis, fault injection, and physical decapsulation.

If an adversary discovers a zero-day exploit and simultaneously compromises millions of ECUs via over-the-air (OTA) network connections, the Sybil issuance rate remains strictly bounded by the untouched SE gate on each vehicle. To bypass this, an adversary must possess multiple genuine SEs or execute highly sophisticated, per-vehicle physical attacks. By reducing a scalable software vulnerability to a non-scalable hardware threat, V-PASS provides a deployable security guarantee.

5 Preliminaries

Before detailing the construction of V-PASS, we define the core cryptographic building blocks employed in our architecture.

5.1 Bilinear Pairings and the q-SDH Assumption

A bilinear pairing is a mapping $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, where $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$ are cyclic groups of prime order p. Let g_1 and g_2 be generators of $\mathbb{G}_1$ and $\mathbb{G}_2$. The pairing operator $\hat{e}$ satisfies: 1) **Bilinearity:** For all $u \in \mathbb{G}_1, v \in \mathbb{G}_2$, and $a, b \in \mathbb{Z}_p$, $\hat{e}(u^a, v^b) = \hat{e}(u, v)^{ab}$. 2) **Non-degeneracy:** $\hat{e}(g_1, g_2) \neq 1$. 3) **Computability:** There exists an efficient algorithm to compute $\hat{e}$.

The security of the EC used in V-PASS relies on the **q-Strong Diffie-Hellman (q-SDH) Assumption**: given a $(q+2)$-tuple $(g_1, g_2, g_2^{\gamma}, g_2^{(\gamma^2)}, \ldots, g_2^{(\gamma^q)})$, it is computationally infeasible for any probabilistic polynomial-time (PPT) adversary to output a pair $(c, g_1^{\frac{1}{\gamma+c}})$ where $c \in \mathbb{Z}_p$. We stress that the security parameter q here is unrelated to the SE-group order q_{se}. Additionally, we define $\mathsf{H}_{\mathbb{G}_1}$ as a hash-to-group function that maps arbitrary bitstrings to elements in $\mathbb{G}_1$; in our security proofs, $\mathsf{H}_{\mathbb{G}_1}$ is modeled as being indifferentiable from a random oracle.

5.2 BBS+ Signatures

BBS+ [7, 12] is a pairing-based signature scheme enabling multi-message signing and highly efficient zero-knowledge proofs of knowledge (ZKPoK).

A BBS+ signature takes the form $\sigma = (A, \epsilon, s_e) \in \mathbb{G}_1 \times \mathbb{Z}_p \times \mathbb{Z}_p$. Let $pk_{EA} \in \mathbb{G}_2$ denote the issuer public key. The verification equation checks: $\hat{e}(A, pk_{EA} \cdot g_2^{\epsilon}) \stackrel{?}{=} \hat{e}(g_1 \cdot h_0^{s_e} \prod_{i=1}^{L} h_i^{m_i}, g_2)$, where $h_0, \ldots, h_L \in \mathbb{G}_1$ are public bases. In V-PASS, our construction builds upon the property that a prover can construct an efficient, non-interactive zero-knowledge proof showing that σ is valid on $\mathbf{m}$ without revealing A, ϵ, s_e, or any undisclosed attribute m_j.

5.3 Zero-Knowledge Proofs

To enable the ECU to prove complex authorization claims without revealing sensitive identifiers, we utilize Non-interactive zero-knowledge proofs (NIZK) based on Σ-protocols [6]. A Σ-protocol is a three-move interactive proof consisting of a commitment from the prover, a challenge from the verifier, and a response from the

prover. In V-PASS, we convert these interactive protocols into non-interactive proofs using the **Fiat-Shamir heuristic**. For a statement s and a witness w, the prover generates a proof $\pi = (c, s_r)$ by computing the challenge $c = \mathsf{H}(s, \mathrm{com})$, where com is the initial protocol commitment. We use the Camenisch-Stadler notation to denote a zero-knowledge proof of knowledge:

$$\pi = \mathrm{ZKPoK}\{(\alpha, \beta, \dots) : \mathcal{R}(\alpha, \beta, \dots, \mathrm{public\_params})\},$$

which proves the knowledge of secret witnesses $(\alpha, \beta, \dots)$ satisfying the relations $\mathcal{R}$ relative to the public parameters. This abstraction allows the ECU to bind multiple algebraic claims (e.g., credential possession and ciphertext correctness) into a single, unforgeable authentication object.

5.4 ElGamal Encryption

To provide conditionally authorized traceability, we utilize ElGamal encryption as a foundational primitive. To achieve algebraic alignment with the EC, we instantiate this primitive over the pairing-friendly group $\mathbb{G}_1$.

Given a public key $pk \in \mathbb{G}_1$ and a message $M \in \mathbb{G}_1$, the encryption algorithm chooses a random scalar $r \leftarrow \mathbb{Z}_p$ and outputs the ciphertext $E = (C_1, C_2)$:

$$C_1 = g_1^r, \quad C_2 = pk^r \cdot M$$

Under the **DDH assumption** in $\mathbb{G}_1$, the ElGamal ciphertexts ensure semantic security (IND-CPA), guaranteeing that observers cannot distinguish between encryptions of different identities.

6 Protocol Construction

We now present a concrete protocol construction that realizes the split-trust architecture. To satisfy the conflicting requirements of high-frequency broadcasting and complex cryptographic authorization, our protocol decouples authentication into two distinct phases: a low-rate *window authorization* executed jointly by the ECU and SE, and a high-rate *payload authentication* executed by the ECU through the SE signing interface.

6.1 Cryptographic Domains and Algebraic Alignment

To avoid expensive cross-curve proof machinery, V-PASS uses *algebraic alignment* across two domains:

Pairing-friendly domain (ECU-side). We instantiate a pairing-friendly elliptic curve hosting groups $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$ of prime order p, with generator $g_1 \in \mathbb{G}_1$. To avoid cross-domain zero-knowledge proofs, *both* the enrollment credential framework and the TA's ElGamal tracing encryption are defined over this domain. Thus, the ECU can construct efficient Σ-protocols over the shared scalar field $\mathbb{Z}_p$, and the TA uses an ElGamal keypair (pk_T, sk_T) in $\mathbb{G}_1$.

Pairing-free domain (SE-side). To accommodate the limited computational capacity of typical automotive SEs, pseudonym key derivation and SE-issued attestations are strictly confined to a standard prime-order elliptic curve $\mathbb{G}_{se}$ (e.g., NIST P-256) of order q_{se} with generator g_{se}. Payload signing (e.g., ECDSA) also occurs in this lightweight domain.

The protocol uses domain-separated hash functions and two credentials: an **EC** (cred) issued by the EA in the pairing-friendly domain, which embeds the hidden identity scalar m_{id} and enables ECU-side pseudonym self-provisioning; and an SE *Attestation Credential* ($\mathrm{cred}_{\mathsf{SE}}$) pre-provisioned by the SE manufacturer, which enables the SE to emit verifiable attestations (σ_{SE}) that it enforced the window-gating logic. We model SE attestation as an **Anonymous Signature Engine (ASE)** with algorithms (Setup, Join, Sign, Verify). We require that Sign produces publicly verifiable attestations under a group/public verification key gpk, while not revealing a stable per-device identifier across signatures (unless explicitly enabled). This is naturally instantiated by anonymous device attestation primitives such as DAA and ECDAA [12, 34]. The SE pseudonym derivation function is modeled as a pseudorandom function $\mathsf{PRF}_s : \{0,1\}^* \to \{0,1\}^\lambda$ keyed by secret s.

6.2 Setup and Enrollment

During system initialization, the trusted authorities publish the public parameters:

$$\mathsf{pp} := (\mathsf{pp}_{\mathsf{cred}}, (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, \hat{e}), (\mathbb{G}_{se}, q_{se}, g_{se}), \\ pk_T, pk_{EA}, gpk, \mathsf{H}_{\mathsf{id}}, \mathsf{H}_{\mathsf{ctx}}, \mathsf{H}_{\mathsf{msg}}, \mathsf{H}_{\mathsf{toZq}}, \mathsf{H}_{\mathbb{G}_1}, \mathsf{H}_{FS}),$$

where $\mathsf{pp}_{\mathsf{cred}}$ contains the base EC public parameters and credential-verification parameters (including public bases for BBS+ and proof parameters for Cred.Prove/Cred.Verify). The EA issuer public key is $pk_{EA} \in \mathbb{G}_2$, while the corresponding secret key sk_{EA} is kept offline by the EA. Here gpk is the SE anonymous attestation verification key. We utilize several domain-separated cryptographic hash functions:

(1) **Hash-to-field functions:** $\mathsf{H}_{\mathsf{id}} : \{0,1\}^* \to \mathbb{Z}_p$ and $\mathsf{H}_{\mathsf{toZq}} : \{0,1\}^* \to \mathbb{Z}_{q_{se}}$ modeled as hash-to-field random oracles for identity mapping and SE-side pseudonym scalar derivation.
(2) **Hash-to-group function:** $\mathsf{H}_{\mathbb{G}_1} : \{0,1\}^* \to \mathbb{G}_1$ for deriving revocation bases.
(3) **Bitstring hash functions:** $\mathsf{H}_{\mathsf{ctx}}, \mathsf{H}_{\mathsf{msg}}, \mathsf{H}_{FS} : \{0,1\}^* \to \{0,1\}^\lambda$ with domain separation, used for context binding and payload hashing, as well as for Fiat–Shamir transcript hashing. In our security analysis, H_{FS} is modeled as a random oracle to justify Fiat–Shamir soundness and zero-knowledge in the standard ROM setting.

For each registered vehicle, the EA first derives a long-term identity scalar:

$$m_{\mathsf{id}} := \mathsf{H}_{\mathsf{id}}(\texttt{"id"} \,\|\, \mathsf{id}) \in \mathbb{Z}_p.$$

The EA then executes the **Enrollment Procedure**:

(1) It verifies the vehicle's real-world identity and confirms its public policy attributes.
(2) It issues the EC by signing an attribute vector $\mathbf{m}$ that includes m_{id} and any policy attributes: $\mathrm{cred} \leftarrow \mathrm{Cred.Issue}(sk_{EA}, \mathbf{m})$. Concretely, cred is a BBS+ signature on $\mathbf{m}$.

Inside the vehicle, the SE is initialized by sampling a master PRF secret s and initializing an empty state dictionary $\mathsf{lastW}[\cdot]$. The SE also holds an attestation credential $\mathrm{cred}_{\mathsf{SE}}$ obtained via the ASE join/provisioning procedure (typically at manufacturing time), which enables ASE.Sign.

The master PRF secret s is freshly generated during each enrollment procedure; therefore, pseudonyms derived from different enrollments are independent even when window indices and contexts coincide. The tuple $(s, \text{lastW}, \text{cred}_{\text{SE}})$ is anchored in the SE's secure storage.

SE Non-Simulability via Per-Window Binding: To preserve cross-window unlinkability, V-PASS does not reveal a stable, per-enrollment hardware identifier in WObj_w. Instead, during window authorization, the ECU binds its pairing-based NIZK proof to the specific, per-window SE attestation signature σ_{SE}.

This prevents a compromised ECU from generating an accepted WObj_w using a *pure software* simulation in lieu of the SE: without access to a genuine SE that can produce a valid attestation signature for window w, verification fails. However, this per-window binding alone does not prevent an extracted EC from being exercised using a different legitimate SE; achieving such strong credential-to-hardware anti-cloning together with cross-window unlinkability requires additional mechanisms beyond our base construction.

6.3 Pseudonym Credential Generation and Presentation

For each time window, the ECU must demonstrate possession of a valid EC without revealing the vehicle's long-term identity. Let $\mathbf{m}$ denote the EC attribute vector containing m_{id} at a hidden index j. Because V2X networks often require Attribute-Based Access Control (e.g., proving emergency vehicle status or a specific emission rating for tolling), the vector $\mathbf{m}$ can simultaneously contain these operation-critical attributes.

During the generation of the self-provisioned pseudonym credential, the ECU cryptographically proves possession of the EC while *selectively disclosing* only the required subset of auxiliary attributes $\mathbf{m}_{reveal} \subset \mathbf{m}$, leaving m_{id} entirely hidden securely in zero-knowledge:

$$\mathcal{L}_{\text{cred}} = \Big\{(\text{pp}_{\text{cred}}, pk_{EA}, \text{ctx}, \text{aux}, \mathbf{m}_{reveal}) \mid \exists(\sigma, \mathbf{m}) : \text{Cred.Verify}(pk_{EA}, \sigma, \mathbf{m}) = 1 \wedge \mathbf{m}[j] = m_{\text{id}}\Big\}. \tag{1}$$

Crucially, the auxiliary input aux binds this credential presentation to the exact public transcript hashed into the unified Fiat–Shamir challenge (Eq. 5 and Appendix C), preventing replay across windows and operational contexts. The ECU executes the presentation:

$$\pi_{\text{cred}} \leftarrow \text{Cred.Prove}(\text{pp}_{\text{cred}}, pk_{EA}, \sigma, \mathbf{m}; \text{aux}, \mathbf{m}_{reveal}).$$

6.4 Window Authorization in SE: Solving Multi-Domain Lockout

Let Δ denote the globally defined window duration. To support multi-service environments (e.g., concurrent V2V safety broadcasting and V2I tolling) without triggering service lockouts, the context ctx is uniquely bound to the specific domain:

$$\text{ctx} := \text{H}_{\text{ctx}}(\texttt{"ctx"} \parallel \text{domain} \parallel \text{policy\_version}).$$

The window duration Δ is a policy parameter. Smaller Δ reduces the maximum time horizon over which transcripts can be linked within a window, at the cost of more frequent window authorization and reduced communication amortization (Section 8). Upon an authorization request for ctx, the SE executes DeriveWindowNym(ctx). It obtains a trusted timestamp/counter to derive the window w and checks that $w > \text{lastW}[\text{ctx}]$, updating the state if valid. Receivers enforce freshness (e.g., within-window tolerances) and accept only window indices consistent with local time; this prevents an ECU from bypassing rate-gating by selecting arbitrary window labels.

This multidimensional rate-gate allows a vehicle to participate in orthogonal V2X services simultaneously, while rigorously enforcing a strict limit of one pseudonym per service per time window. The SE then derives the window-specific pseudonym keying material in $\mathbb{Z}_{q_{se}}$:

$$\begin{aligned} \text{seed}_w &:= \text{PRF}_s(\texttt{"nym"} \parallel w \parallel \text{ctx}), \\ \alpha_w &:= \text{H}_{\text{toZq}}(\texttt{"nym\_scalar"} \parallel \text{seed}_w), \\ \text{nym}_w &:= g_{se}^{\alpha_w}. \end{aligned} \tag{2}$$

If the hash-to-scalar result yields $\alpha_w = 0$ (an event occurring with negligible probability $1/q_{se}$), the SE re-derives seed_w by appending a domain-separation counter (e.g., $\text{seed}_w := \text{PRF}_s(\texttt{"nym"} \parallel w \parallel \text{ctx} \parallel 1)$) and repeats the derivation to ensure a valid non-zero secret key. Finally, the SE computes an anonymous hardware attestation to explicitly endorse this derived pseudonym and its associated rate-limiting context:

$$\sigma_{\text{SE}} \leftarrow \text{ASE.Sign}(\text{cred}_{\text{SE}}, \texttt{"rate"} \parallel w \parallel \text{ctx} \parallel \text{nym}_w).$$

The SE retains the private pseudonym key α_w internally and returns only an opaque key handle handle_w together with the authorization outputs: $(w, \text{nym}_w, \sigma_{\text{SE}}, \text{handle}_w)$. The ECU cannot access or export α_w. All pseudonym signing operations must be requested through the SE interface using handle_w.

6.5 Window Object Integration on ECU

Given the SE's pairing-free output and the EA's pairing-friendly EC, the ECU constructs a unified, reusable authorization object for window w.

First, the ECU generates the traceability ciphertext over $\mathbb{G}_1$. It forms the identity trace tag $Y_{\text{id}} := g_1^{m_{\text{id}}}$, protecting against static identifier leakage, and encrypts it under the TA's public key using fresh randomness $r \leftarrow \mathbb{Z}_p$:

$$E = (C_1, C_2) = \text{Enc}_{pk_T}(Y_{\text{id}}; r) = (g_1^r, pk_T^r \cdot g_1^{m_{\text{id}}}).$$

Next, the ECU synthesizes a composed NIZK proof π, which simultaneously demonstrates: (i) possession of a valid EC, (ii) correctness of the ElGamal ciphertext $E = (C_1, C_2)$ encrypting identity $Y_{\text{id}} = g_1^{m_{\text{id}}}$, and (iii) validity of the revocation tag $R_w = \text{H}_{\mathbb{G}_1}(\texttt{"rev"} \parallel w)^{m_{\text{id}}}$. At a high level:

$$\begin{aligned} \pi = \text{ZKPoK}\{(m_{\text{id}}, r, \text{wit}_{\text{cred}}) : \\ \text{Cred.Rel}(\text{pp}_{\text{cred}}, pk_{EA}; \text{wit}_{\text{cred}}, m_{\text{id}}) = 1 \\ \wedge\, C_1 = g_1^r \wedge C_2 = pk_T^r \cdot g_1^{m_{\text{id}}} \\ \wedge\, R_w = \text{H}_{\mathbb{G}_1}(\texttt{"rev"} \parallel w)^{m_{\text{id}}}\}. \end{aligned} \tag{3}$$

where wit_{cred} denotes the witness used by the EC proof subsystem, and $\text{Cred.Rel}(\cdot)$ is the credential-validity relation enforced by that subsystem (Appendix A).

6.5.1 Cross-Domain Fiat-Shamir Challenge. By designing TA tracing and revocation statements in the same scalar field $\mathbb{Z}_p$, the composed proof shares witness m_{id} across sub-statements. The Fiat–Shamir challenge is derived from the canonical transcript:

$$c \leftarrow \mathsf{H}_{FS}(\mathsf{pp}, \mathsf{ctx}, w, \mathsf{nym}_w, \sigma_{\mathsf{SE}}, C_1, C_2, R_w, T_{\mathsf{cred}}, T_{C_1}, T_{C_2}, T_R).$$

where $(T_{\mathsf{cred}}, T_{C_1}, T_{C_2}, T_R)$ are the forward commitments of the composed Σ-protocol (defined formally in Appendix A). The scalar challenge used by the ECU's pairing-friendly Σ-protocol is $c_p = \mathsf{ToField}_p(c)$, where $\mathsf{ToField}_p(x) = \mathsf{OS2IP}(x) \bmod p$ and OS2IP is the octet-string-to-integer conversion.

To avoid ambiguity in transcript reconstruction, the verifier reconstructs the exact same byte string using a canonical encoding and explicit domain separation (Appendix C). This yields two implementation-security properties: (1) *binding integrity*—the proof cannot be transplanted to a different $(w, \mathsf{ctx}, \mathsf{nym}_w, \sigma_{\mathsf{SE}})$ tuple; and (2) *statement completeness*—all publicly verified objects are included in challenge derivation.

This binds the ECU proof to the specific per-window SE attestation without cross-curve zero-knowledge statements. The ECU packages these components into:

$$\mathsf{WObj}_w := (\mathsf{nym}_w, \sigma_{\mathsf{SE}}, E, R_w, \pi).$$

This WObj_w is computed once per window, substantially reducing pairing-related cost in the high-frequency broadcast path.

6.6 Payload Authentication and Stateless Verification

For every high-frequency V2X payload P transmitted during window w, the ECU only performs a lightweight operation in the $\mathbb{G}_{se}$ domain. The message hash inherently binds to the pseudonym and context to thwart subtle cross-domain replay attacks:

$$\begin{aligned} \mu &:= \mathsf{H}_{\mathsf{msg}}(\texttt{"msg"}, w, \mathsf{ctx}, \mathsf{nym}_w, P), \\ \sigma_P &\leftarrow \mathsf{SE.Sign}(\mathsf{handle}_w, \mu). \end{aligned} \tag{4}$$

The vehicle transmits the tuple $(w, P, \sigma_P, \mathsf{WObj}_w)$. To optimize bandwidth, the sender may periodically attach WObj_w (e.g., once every k beacons) rather than on every payload, assuming temporal locality of verifiers.

When a packet carries WObj_w, verification is self-contained. In the amortized mode where WObj_w is attached only periodically, receivers may cache a validated WObj_w for the duration of the window to authenticate intermediate packets; our security arguments do not rely on such caching.

Upon receipt, an ephemeral verifier (such as a neighboring vehicle or RSU) recomputes ctx and performs four local checks (see Appendix B for a concrete verifier procedure):

(1) **Payload Integrity:** ECDSA signature (σ_P) validity under nym_w.
(2) **Hardware Rate Limit:** Anonymous attestation (σ_{SE}) valid for $(w, \mathsf{ctx}, \mathsf{nym}_w)$ under gpk.
(3) **Revocation Check:** Verify $R_w \notin \mathsf{Blocklist}_w$.
(4) **Authorization & Accountability:** NIZK validity for π.

Because all policy-enforcement state is sender-local in the SE, receivers do not need to maintain long-lived per-sender issuance state; any caching is optional and bounded to a time window.

6.7 Conditional Deanonymization and Revocation

In the event of observed misbehavior, an authorized incident report containing a valid WObj_w is forwarded to the TA. The TA validates the NIZK proof π to guarantee ciphertext well-formedness, and then executes the ElGamal decryption in $\mathbb{G}_1$:

$$Y_{\mathsf{id}} = \mathrm{Dec}_{sk_T}(C_1, C_2) = C_2 \cdot (C_1^{sk_T})^{-1} = g_1^{m_{\mathsf{id}}}.$$

Against the recovered Y_{id}, the TA performs a lookup in the secure enrollment database to reveal the long-term identity m_{id}. This avoids expensive discrete-logarithm solving and permits immediate, granular revocation of individual machines.

VLR revocation tags (R_w) are broadcast alongside π; verifiers then perform a stateless membership test $R_w \in \mathsf{Blocklist}_w$.

$$R_w := \mathsf{H}_{\mathbb{G}_1}(\texttt{"rev"} \parallel w)^{m_{\mathsf{id}}} \in \mathbb{G}_1$$

Because R_w does not include ctx, revocation is *global* across all service contexts: once an enrollment is revoked, its packets are rejected regardless of the service being used in that window. This also implies that transcripts within the same window are linkable via R_w across contexts; this is by design and is bounded in time by Δ. Within the NIZK π, the ECU proves that the base scalar m_{id} used to compute R_w is identical to the hidden enrollment identity.

Once a vehicle is classified as malicious, the TA (or the revocation infrastructure) retrieves the corresponding scalar m_{id}^* from enrollment records indexed by Y_{id} (or recomputes it from the enrolled identity string) and then computes the tags R_w^* for current and future windows to distribute as a compact blocklist to verifiers.

Under DDH in $\mathbb{G}_1$ and random-oracle modeling of $\mathsf{H}_{\mathbb{G}_1}$, tags from different windows are computationally unlinkable for parties without m_{id} or TA capabilities (Section 7).

Because R_w is deterministic for fixed (m_{id}, w), all accepted packets in window w from that enrollment map to the same tag. Verifiers therefore perform a stateless per-packet membership test $R_w \in \mathsf{Blocklist}_w$.

7 Formal Security Analysis

We analyze V-PASS under the threat model of Section 3.2. The analysis uses standard assumptions: (i) unforgeability of the enrollment credential system (e.g., BBS+ under q-SDH), (ii) soundness and zero-knowledge of the composed Fiat–Shamir proof in the random-oracle model, (iii) unforgeability of SE attestation outputs under chosen-message attacks, (iv) SE monotonic-state integrity, and (v) DDH security in $\mathbb{G}_1$ for ElGamal and revocation-tag unlinkability arguments. Consistent with the protocol composition, credential soundness underwrites authorization, SE state and attestation underwrite rate limiting and non-simulability, and ElGamal/ZK underwrite transcript privacy for honest vehicles.

Definition 7.1 (Authorization Correctness). A V2X pseudonym self-provisioning protocol satisfies Authorization Correctness if for any PPT adversary $\mathcal{A}$ lacking a legitimately issued EC from the EA, the probability of producing a valid protocol transcript $T = (w, P, \sigma_P, \mathsf{WObj}_w)$ is negligible:

$$\Pr\left[\mathsf{Verify}(w, P, \sigma_P, \mathsf{WObj}_w) = 1\right] \le \mathsf{negl}(\lambda)$$

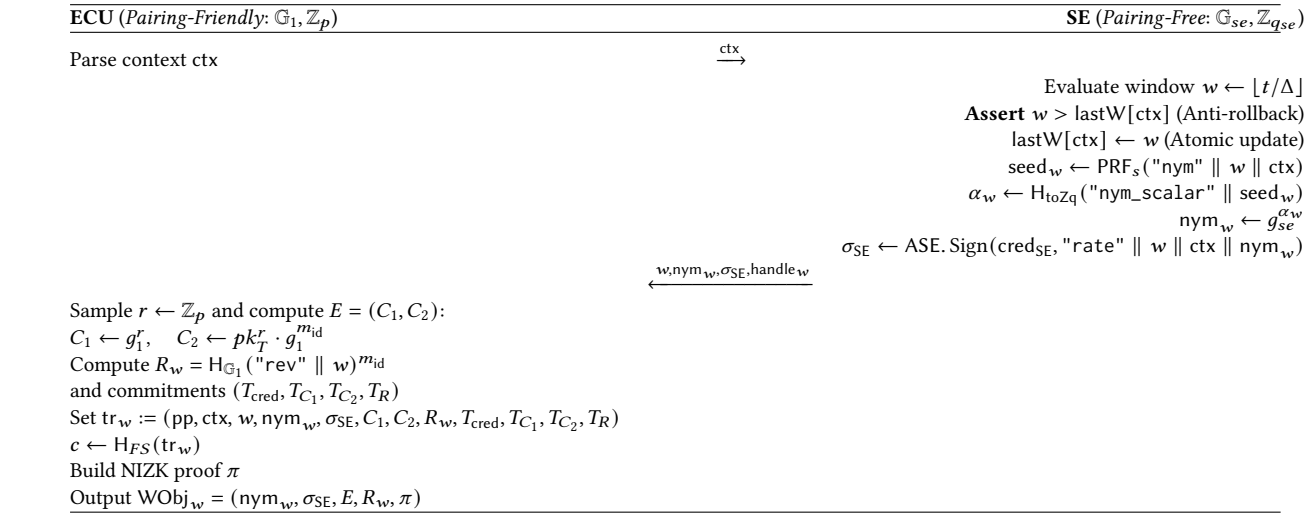


Figure 2: The V-PASS Interactive Window Authorization Protocol. This split-trust flow generates the unified WObj_w at the start of epoch w. The SE acts as the immutable rate-limiting root, while ZKP generation is offloaded to the ECU.

Any cryptographically accepted authentication object must demonstrate possession of a valid cred issued by the EA.

Theorem 7.2. *V-PASS provides Authorization Correctness under the assumption of enrollment-credential unforgeability and knowledge soundness of the Fiat–Shamir NIZK in the Random Oracle Model (ROM).*

Proof Sketch. Suppose an unenrolled adversary outputs an accepted transcript. Verification accepts only if π proves possession of a valid enrollment credential and consistency of the ciphertext/tag relations. By Fiat–Shamir knowledge soundness (via forking-style extraction in ROM), acceptance implies extraction of a valid credential witness, contradicting credential unforgeability except with negligible probability. □

Definition 7.3 (Hardware-Enforced Ex-Ante Sybil Resistance). A protocol provides single-SE ex-ante Sybil resistance if, for a fixed genuine SE instance, any PPT adversary controlling the ECU can obtain at most one accepted authorization for a fixed (ctx, w), except with negligible probability.

Theorem 7.4. *V-PASS provides single-SE ex-ante Sybil resistance under SE monotonic-state integrity and attestation unforgeability.*

Proof Sketch. The SE updates lastW[ctx] monotonically and only signs if $w >$ lastW[ctx]. Therefore a second accepted authorization for the same (ctx, w) would require either violating SE state integrity or producing a fresh valid attestation without SE signing access, contradicting attestation unforgeability. □

Definition 7.5 (SE Non-Simulability). V-PASS satisfies SE non-simulability if any PPT adversary without access to a genuine SE signing oracle cannot produce an accepted transcript containing a valid attestation on $\text{"rate"} \parallel w \parallel \text{ctx} \parallel \text{nym}_w$, except with negligible probability.

Theorem 7.6. *V-PASS satisfies SE non-simulability assuming attestation unforgeability.*

Proof Sketch. Acceptance requires a valid attestation under *gpk* on the exact per-window message included in the Fiat–Shamir transcript. Without SE access, producing such an attestation is precisely an attestation-forgery event. □

Definition 7.7 (Computational Anonymity of the Enrollment Identity). V-PASS satisfies computational anonymity of the enrollment identity if any PPT adversary without TA decryption capability learns no efficiently testable identifier for m_{id} from accepted transcripts beyond public side information.

Theorem 7.8. *For honest (non-compromised) vehicles, V-PASS provides computational anonymity against non-TA observers under DDH in $\mathbb{G}_1$ and zero-knowledge of π in ROM.*

Proof Sketch. Identity-related public values are $E = (C_1, C_2)$ and R_w. ElGamal rerandomization in $\mathbb{G}_1$ gives IND-CPA security under DDH, so E does not reveal m_{id} to non-TA parties. For tags, $R_w = \text{H}_{\mathbb{G}_1}(\text{"rev"} \parallel w)^{m_{\text{id}}}$ uses window-dependent bases; with random-oracle hash-to-group modeling and DDH hardness, tags across distinct windows are computationally unlinkable without m_{id}. Finally, π is zero-knowledge and does not reveal the witness. □

7.1 Hardware Boundary and Side-Channel Considerations

The security of V-PASS's rate-limiting relies on the isolation of the SE's monotonic state and its attestation key. While we treat the SE as a black-box root of trust, production automotive HSMs are typically certified to high evaluation assurance levels (e.g., CC EAL5+) and include hardware-level countermeasures against power analysis and electromagnetic side-channels.

Even if an adversary achieves partial side-channel leakage on a specific vehicle's SE, the impact is strictly contained: the adversary could only spoof authorizations for *that specific vehicle*. Unlike pure software-based systems where a single discovered exploit (e.g., a

buffer overflow) can be weaponized into a global Sybil infrastructure with zero marginal cost, V-PASS forces an adversary to scale physical attacks to each individual hardware target. This raises the economic bar for large-scale Sybil attacks from software scripts to per-vehicle hardware complexity, which is consistent with the security objectives of production vehicular safety systems.

An expanded decomposition-style argument for privacy is given in Appendix D. Consistent with the threat model (Section 3.2), we claim anonymity only for honest vehicles; if ECU secrets are extracted (including m_{id}), the adversary can recompute its own historical tags. We also do not claim unlinkability *within* a window: by construction, packets in the same window share nym_w and the global revocation tag R_w, which enables within-window linkage (including across service contexts). This linkage is bounded in time by the policy-chosen window duration Δ.

8 Implementation and Evaluation

Our evaluation addresses three questions about practical deployability: 1) **Computational Feasibility (RQ1):** Does the split-trust architecture meet 10 Hz real-time constraints on automotive-grade hardware? 2) **Communication Efficiency (RQ2):** How does amortized overhead of WObj_w compare to VPKI? 3) **Security Enforcement (RQ3):** Does SE-enforced window gating block repeated same-window authorization attempts from a compromised ECU?

8.1 Prototyping Setup

We implemented V-PASS on a **Raspberry Pi 5** (Cortex-A76) as an ECU prototype, representative of Cortex-A–class CPUs used in modern automotive SoCs (e.g., Qualcomm Snapdragon Ride).

(1) **SE:** Monotonic state, PRF-derived keys, and SECP256R1 signing.
(2) **ECU Functionality:** Pairing-based credential proving and transcript binding using a Rust-backed crypto implementation (`bls12_381`, `bbs` [1]).
(3) **Interconnect Model:** ECU–SE interaction with serial-bus latency emulation (mean 254 μs).

8.2 Methodology and Reporting Conventions

We report mean and 99th-percentile latency for all critical operations. For setup-path decomposition, we separate: (i) SE-side window authorization primitives, (ii) ECU-side pairing-heavy proof generation, and (iii) end-to-end WObj_w build latency.

For communication analysis, we consider two accounting views:

- **Protocol payload size:** byte size of cryptographic fields, used for protocol-level comparison.
- **Encoded message size:** transport-level size including serialization overhead, used for implementation realism checks.

Unless otherwise stated, figures and break-even points are computed using protocol payload size to avoid conflating protocol design with encoding choices.

For fairness against the baseline, we evaluate a per-packet VPKI path with payload-signature and certificate verification under the same software environment. We then compare V-PASS in amortized form over k packets that reuse one window object.

Phase	Operation	Mean (ms)	99th (ms)
SE Side	Secure-Time Read + Counter Update	0.005	0.009
	PRF / Pseudonym Derivation	0.122	0.141
	SE Attestation Sign	0.082	0.093
ECU Side	Composed NIZK Proof π	47.868	48.873
	ElGamal Trace Ciphertext	8.119	8.198
	Commitments and Transcript	13.899	14.045
Total Setup	WObj_w Construction	**70.095**	**71.359**
Steady-State	Per-Message Payload Sign	0.126	0.181
	Verifier (Full WObj_w Check)	14.606	14.894

Table 1: Micro-benchmarks on Raspberry Pi 5. Heavy pairing-based ZKPs are isolated in the ECU setup phase, while steady-state signing remains extremely efficient.

8.3 Micro-Benchmarks: Computational Feasibility (RQ1)

We decompose the overhead into the *Authorization Loop* (rare epoch-level setup) and the *Authentication Loop* (high-frequency steady-state signing).

Window Setup (Low-Rate): Table 1 shows that SE-side authorization primitives are sub-millisecond, while ECU-side pairing-based proving dominates setup. End-to-end WObj_w construction is **70.1 ms** on average, and is executed once per window rather than once per beacon.

Message Signing (High-Rate): In steady state, payload signing through an SE handle costs **0.126 ms** per message, supporting 10 Hz beaconing with substantial headroom.

8.4 Communication Efficiency (RQ2)

In V2X ad-hoc networks, bandwidth is a scarce resource. Traditional VPKI carries a certificate and payload signature per packet, while V-PASS amortizes a larger WObj_w across multiple packets in the same window.

In our amortization model, we use fixed-width signatures (64B) and a measured raw WObj_w size of about 1.06KB, yielding a VPKI authentication baseline of 264B/packet. We define k as the number of beacons sent within a single window, and compute amortized authentication overhead as $|\sigma_P| + |\mathrm{WObj}_w|/k$. As shown in Figure 4, break-even occurs at about $k = 6$ packets. For longer contacts (e.g., $k = 100$), the authentication-overhead reduction is substantial. Under the paper's payload-inclusive accounting, the total per-packet bandwidth reduction at large k is about 43%.

This trend is robust to moderate parameter shifts. Increasing certificate size or reducing window-object size lowers the break-even k; conversely, shorter contact durations or highly bursty one-shot contacts reduce amortization gains. The figure should therefore be interpreted as a contact-duration sensitivity curve rather than a single-point claim. Finally, Δ influences the *maximum* achievable amortization because k is upper-bounded by the beacon rate times the window duration. Smaller windows improve privacy by shortening the within-window linkability horizon, but also reduce the opportunity to amortize WObj_w across many packets.

8.5 Memory and Verifier Scalability

V-PASS significantly reduces the local storage burden on vehicular platforms compared to traditional VPKI systems.

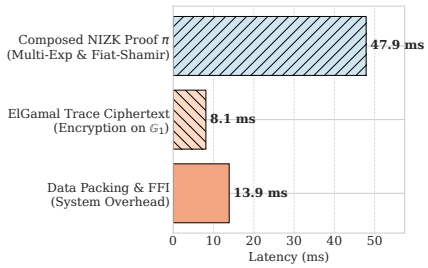


Figure 3: Latency breakdown of ECU-side WObj_w construction. Pairing-friendly proof generation dominates setup cost (≈ 48 ms including binding proof).

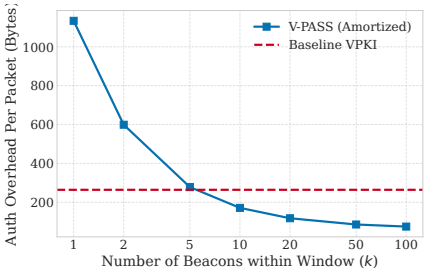


Figure 4: Communication-cost amortization. V-PASS becomes smaller than the VPKI baseline after approximately six beacons per window.

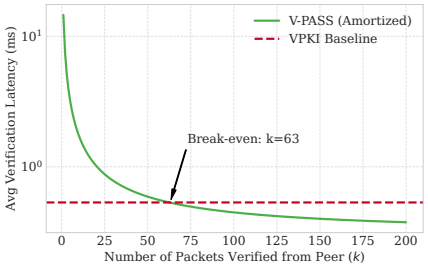


Figure 5: Amortized verification latency. V-PASS becomes more efficient than the VPKI baseline after approximately 63 packets from the same peer.

Storage Complexity. In a standard VPKI model, to maintain privacy for 30 days without network connectivity, a vehicle might pre-fetch 20 certificates per day per context. For 5 active contexts, this requires storing 3,000 certificates (each ≈ 500–800 bytes), totaling over 2 MB of persistent storage. In contrast, V-PASS requires only the long-lived EC (≈ 300 bytes) and the current active window objects. This represents a **1750× reduction** in long-term pseudonym storage.

Communication Bandwidth and Amortization. A key trade-off in V-PASS is the size of the WObj_w, which includes a BBS+ based NIZK and traceability ciphertexts. On the BLS12-381 curve, a WObj_w occupies approximately 864 bytes. While this exceeds the size of a standard ECDSA signature, our *amortization strategy* mitigates this cost. By broadcasting WObj_w only once every k packets, the effective per-message overhead is $(\mathrm{size}(\sigma_P) + \mathrm{size}(\mathrm{WObj}_w)/k)$. For a 10 Hz safety beacon ($k = 100$ per 10 seconds), the additional overhead drops to less than 9 bytes per packet. Figure 5 illustrates that for $k \geq 10$, the total bandwidth utilized per window is effectively comparable to standard certificate-rotation models, while providing the added benefit of ex-ante hardware gating.

Verifier Performance: When a packet carries WObj_w, full verification (ASE attestation, NIZK, and payload signature) costs 14.6 ms on average in our prototype. In the amortized mode where WObj_w is attached once every k packets, a verifier may cache the validated (nym_w, R_w) for the duration of the window; subsequent packets then require only the payload-signature check, costing 0.30 ms. Figure 5 shows a break-even point at $k \approx 63$ packets, i.e., about 6.3 seconds at 10 Hz.

8.6 Security Enforcement: Sybil Defense (RQ3)

To validate Goal 2, we simulated a compromised ECU requesting 50 pseudonyms for one (ctx, w) pair. The SE granted one request and rejected the remaining 49 by enforcing $w > \mathrm{lastW}[\mathrm{ctx}]$. This is consistent with our single-SE ex-ante rate-limiting claim against software adversaries.

Evaluation scope. Our evaluation targets three questions aligned with our design goals and threat model: (i) feasibility of the two-loop split on ECU-class hardware, (ii) communication amortization under multi-packet contacts, and (iii) behavior of sender-side monotonic gating under ECU compromise.

Because our SE functionality is emulated and our interconnect is modeled, absolute latency values will depend on the concrete production SE and vehicle integration. We therefore emphasize the decomposition and amortization trends that drive deployability (setup versus steady-state costs) rather than claiming silicon-specific numbers.

9 Conclusion

This paper presented V-PASS, a hardware-assisted architecture for privacy-preserving V2X authentication that eliminates large-scale pseudonym provisioning while preventing uncontrolled Sybil identity generation. By combining local pseudonym derivation with Secure-Element–assisted authorization, V-PASS ensures that at most one valid pseudonym can be derived per window even under ECU software compromise.

Our split-trust design separates lightweight authorization inside the SE from computationally intensive cryptographic operations executed on the ECU, enabling strong privacy guarantees while remaining compatible with constrained automotive hardware. The resulting window object supports stateless receiver verification while preserving unlinkability and conditional traceability.

Acknowledgment

This work was supported in part by the Office of Naval Research under grants N00014-24-1-2730, and the National Science Foundation under grants 2433904, 2433905, 2247560, 2247561, 2442382, 2154929, 2331936, 2332675, 2235232, and 2312447.

References

[1] [n. d.]. bbs - Rust. https://docs.rs/bbs/latest/bbs/.
[2] 2016. SAE J2735: Dedicated Short-Range Communications (DSRC) Message Set Dictionary.
[3] 2022. IEEE Standard for Wireless Access in Vehicular Environments (WAVE) – Certificate Management Interfaces for End Entities. https://standards.ieee.org/ieee/1609.2.1/10728/ Corrigendum: IEEE Std 1609.2.1-2022/Cor 1-2023.
[4] 2025. *Intelligent Transport Systems (ITS); Security; Pre-standardization study on pseudonym change management; Release 2.* Technical Report ETSI TR 103 415 V2.1.1. ETSI.
[5] AutoCrypt Co., Ltd. 2023. V2X PKI Certificates: Butterfly Key Expansion and Implicit Certificates. https://autocrypt.io/v2x-pki-certificates-butterfly-key-expansion-implicit-certificates/.
[6] Manuel Blum, Paul Feldman, and Silvio Micali. 2019. Non-interactive zero-knowledge and its applications. In *Providing sound foundations for cryptography: on the work of Shafi Goldwasser and Silvio Micali.* 329–349.
[7] Dan Boneh, Xavier Boyen, and Hovav Shacham. 2004. Short group signatures. In *Annual international cryptology conference.* Springer, 41–55.

[8] Benedikt Brecht, Dean Therriault, André Weimerskirch, William Whyte, Virendra Kumar, Thorsten Hehn, and Roy Goudy. 2018. A security credential management system for V2X communications. *IEEE Transactions on Intelligent Transportation Systems* 19, 12 (2018), 3850–3871.

[9] Xuyuan Cai, Rui Song, Bin Xie, Qingjun Xiao, and Bin Xiao. 2025. PSP: A Privacy-Preserving Self-certify Pseudonym Protocol for V2X. In *Proceedings of the 20th ACM Asia Conference on Computer and Communications Security*. 651–664.

[10] Zekun Cai and Aiping Xiong. 2023. Understand users' privacy perception and decision of {V2X} communication in connected autonomous vehicles. In *32nd USENIX Security Symposium (USENIX Security 23)*. 2975–2992.

[11] Giorgio Calandriello, Panos Papadimitratos, Jean-Pierre Hubaux, and A. Lioy. 2011. On the Performance of Secure Vehicular Communication Systems. *IEEE Transactions on Dependable and Secure Computing (IEEE TDSC)* (2011).

[12] Jan Camenisch, Manu Drijvers, and Anja Lehmann. 2016. Anonymous attestation using the strong diffie hellman assumption revisited. In *International Conference on Trust and Trustworthy Computing*. Springer, 1–20.

[13] Shan Chang, Yong Qi, Hongzi Zhu, Jizhong Zhao, and Xuemin Shen. 2011. Footprint: Detecting Sybil attacks in urban vehicular networks. *IEEE Transactions on Parallel and Distributed Systems* 23, 6 (2011), 1103–1114.

[14] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno. 2011. Comprehensive experimental analyses of automotive attack surfaces. In *20th USENIX security symposium (USENIX Security 11)*.

[15] Common Criteria Development Board. 2025. Common Criteria for Information Technology Security Evaluation. https://www.commoncriteriaportal.org

[16] ETSI. 2022. ETSI TS 102 941: Intelligent Transport Systems (ITS); Security; Trust and Privacy Management.

[17] European Commission. 2025. Cooperative, connected and automated mobility (CCAM). https://transport.ec.europa.eu/transport-themes/smart-mobility/cooperative-connected-and-automated-mobility-ccam_en.

[18] John Harding and et al. 2014. Vehicle-to-Vehicle Communications: Readiness of V2V Technology for Application.

[19] Christopher Hicks and Flavio D Garcia. 2020. A vehicular DAA scheme for unlinkable ECDSA pseudonyms in V2X. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 460–473.

[20] ISO/SAE. 2021. ISO/SAE 21434:2021 Road vehicles — Cybersecurity engineering.

[21] Martin Kayondo, Junseung You, Eunmin Kim, Jiwon Seo, and Yunheung Paek. 2026. SECV: Securing Connected Vehicles with Hardware Trust Anchors. In *Proceedings of the 33rd Annual Network and Distributed System Security Symposium (NDSS)*.

[22] Salabat Khan, Fei Luo, Zijian Zhang, Mussadiq Abdul Rahim, Mubashir Ahmad, and Kaishun Wu. 2022. Survey on issues and recent advances in vehicular public-key infrastructure (VPKI). *IEEE Communications Surveys & Tutorials* (2022).

[23] Mohammad Khodaei, Andreas Messing, and Panos Papadimitratos. 2017. RHyTHM: A Randomized Hybrid Scheme To Hide in the Mobile Crowd. In *IEEE Vehicular Networking Conference (IEEE VNC)*. Torino, Italy.

[24] Mohammad Khodaei, Hamid Noroozi, and Panos Papadimitratos. 2018. Privacy Preservation through Uniformity. In *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. 279–280.

[25] Mohammad Khodaei, Hamid Noroozi, and Panos Papadimitratos. 2023. SECMACE+: upscaling pseudonymous authentication for large Mobile systems. *IEEE Transactions on Cloud Computing* 11, 3 (2023), 3009–3026.

[26] Mohammad Khodaei and Panos Papadimitratos. 2021. Scalable & Resilient Vehicle-Centric Certificate Revocation List Distribution in Vehicular Communication Systems. *IEEE Transactions on Mobile Computing (IEEE TMC)* 20, 7 (July 2021), 2473–2489.

[27] Charlie Miller and Chris Valasek. 2015. Remote exploitation of an unaltered passenger vehicle. *Black Hat USA* 2015, S 91 (2015), 1–91.

[28] Hamid Noroozi, Mohammad Khodaei, and Panos Papadimitratos. 2018. VPKIaaS: A highly-available and dynamically-scalable vehicular public-key infrastructure. In *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. 302–304.

[29] Jonathan Petit, Florian Schaub, Michael Feiri, and Frank Kargl. 2014. Pseudonym schemes in vehicular networks: A survey. *IEEE communications surveys & tutorials* 17, 1 (2014), 228–255.

[30] Alastair Ruddle et al. 2011. *EVITA: E-safety vehicle intrusion protected applications*. Technical Report. Deliverable D2.3. http://www.evita-project.org/

[31] Roshan Sedar, Charalampos Kalalas, Francisco Vázquez-Gallego, and Jesus Alonso-Zarate. 2022. Reinforcement learning based misbehavior detection in vehicular networks. In *ICC 2022-IEEE International Conference on Communications*.

[32] Md Hasan Shahriar, Mohammad Raashid Ansari, Jean-Philippe Monteuuis, Md Shahedul Haque, Cong Chen, Jonathan Petit, Y Thomas Hou, and Wenjing Lou. 2025. Vehigan: Generative adversarial networks for adversarially robust v2x misbehavior detection systems. *ACM Transactions on Cyber-Physical Systems* (2025).

[33] The Linux Foundation. 2024. Let's Encrypt Case Study. https://www.linuxfoundation.org/resources/case-studies/lets-encrypt.

[34] Trusted Computing Group. 2018. TCG TPM 2.0 Automotive Thin Profile For TPM Family 2.0. https://trustedcomputinggroup.org/resource/tcg-tpm-2-0-library-profile-for-automotive-thin/.

[35] U.S. Department of Transportation. 2019. Secure Credential Management System (SCMS). https://rosap.ntl.bts.gov/view/dot/43635/dot_43635_DS1.pdf.

[36] Bart Van Arem, Cornelie JG Van Driel, and Ruben Visser. 2006. The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Transactions on intelligent transportation systems* 7, 4 (2006), 429–436.

[37] Qianpeng Wang, Deyun Gao, and Du Chen. 2020. Certificate revocation schemes in vehicular networks: A survey. *IEEE Access* 8 (2020), 26223–26234.

[38] William Whyte, André Weimerskirch, Virendra Kumar, and Thorsten Hehn. 2013. A security credential management system for V2V communications. In *2013 IEEE vehicular networking conference*. IEEE, 1–8.

[39] Takahito Yoshizawa, Dave Singelée, Jan Tobias Muehlberg, Stephane Delbruel, Amir Taherkordi, Danny Hughes, and Bart Preneel. 2023. A survey of security and privacy issues in v2x communication systems. *Comput. Surveys* 55, 9 (2023).

A Formal Specification of the ECU Zero-Knowledge Protocol

The V-PASS ECU generates a composed NIZK proof π to simultaneously demonstrate: (i) possession of a valid enrollment credential (EC), (ii) correctness of the ElGamal ciphertext $E = (C_1, C_2)$ encrypting the hidden identity $Y_{\mathsf{id}} = g_1^{m_{\mathsf{id}}}$, and (iii) correctness of the deterministic revocation tag $R_w = \mathsf{H}_{\mathbb{G}_1}(\texttt{"rev"} \parallel w)^{m_{\mathsf{id}}}$.

All algebraic sub-statements are proven in the pairing-friendly group of order p, and a single Fiat–Shamir challenge bitstring $c \in \{0,1\}^\lambda$ is derived over the full public transcript. The scalar challenge used by the Σ-protocol is then $c_p := \mathsf{OS2IP}(c) \bmod p$, where OS2IP is the standard octet-string-to-integer primitive.

A.1 EC Sub-Proof (BBS+ Selective Disclosure)

Let the EC be a BBS+ signature $\sigma_{\mathsf{EC}} = (A, \epsilon, s_e)$ over an attribute vector $\mathbf{m} = (m_1, \ldots, m_L)$ under public bases $(g_1, h_0, \{h_i\}_{i=1}^L)$ and issuer public key $pk_{EA} \in \mathbb{G}_2$. The verification equation is: $\hat{e}(A, pk_{EA} \cdot g_2^{\epsilon}) = \hat{e}(g_1 \cdot h_0^{s_e} \prod_{i=1}^L h_i^{m_i}, g_2)$. Let $m_j = m_{\mathsf{id}}$ be the hidden long-term identity attribute at secret index j. The EC sub-proof π_{cred} is the standard BBS+ proof-of-knowledge with selective disclosure, instantiated as a Σ-protocol and made non-interactive via Fiat–Shamir. We treat its commitment transcript as T_{cred} and its responses as part of π_{cred}.

A.2 ElGamal and Revocation Sub-Proofs (Representation Relations)

Let $E = (C_1, C_2)$ be the ElGamal ciphertext and let R_w be the VLR tag for window w. The ECU proves knowledge of $(m_{\mathsf{id}}, r) \in \mathbb{Z}_p^2$ such that: $C_1 = g_1^r, \quad C_2 = pk_T^r \cdot g_1^{m_{\mathsf{id}}}, \quad R_w = \mathsf{H}_{\mathbb{G}_1}(\texttt{"rev"} \parallel w)^{m_{\mathsf{id}}}$.

This is a standard Σ-protocol for knowledge of a representation with a shared exponent m_{id} across C_2 and R_w. The ECU samples masks $t_r, t_m \leftarrow \mathbb{Z}_p$ and computes forward commitments:

$$\begin{aligned} T_{C_1} &= g_1^{t_r}, \\ T_{C_2} &= pk_T^{t_r} \cdot g_1^{t_m}, \\ T_R &= \mathsf{H}_{\mathbb{G}_1}(\texttt{"rev"} \parallel w)^{t_m}. \end{aligned}$$

It computes the unified Fiat–Shamir challenge bitstring $c \in \{0,1\}^\lambda$:

$$\begin{aligned} c \leftarrow \mathsf{H}_{FS}(&\mathsf{pp}, \mathsf{ctx}, w, \mathsf{nym}_w, \sigma_{\mathsf{SE}}, C_1, C_2, R_w, \\ &T_{\mathsf{cred}}, T_{C_1}, T_{C_2}, T_R) \end{aligned} \tag{5}$$

The ECU derives the scalar challenge $c_p = \mathsf{OS2IP}(c) \bmod p$ to compute responses. The ECU then computes the scalar responses

for the representation sub-proof:

$$z_m = t_m + c_p \cdot m_{\mathsf{id}} \bmod p, \quad z_r = t_r + c_p \cdot r \bmod p.$$

The final proof is output as: $\pi = (\pi_{\mathsf{cred}}, c, T_{C_1}, T_{C_2}, T_R, z_m, z_r)$.
The verifier asserts:

(1) Recompute the challenge hash:

$$c \stackrel{?}{=} \mathsf{H}_{FS}(\mathsf{pp}, \mathsf{ctx}, w, \mathsf{nym}_w, \sigma_{\mathsf{SE}}, C_1, C_2, R_w, T_{\mathsf{cred}}, T_{C_1}, T_{C_2}, T_R).$$

(2) Verify the EC sub-proof π_{cred} according to the credential proof system instantiated in Section 6, using the same transcript binding.
(3) Verify the representation equations:

$$g_1^{z_r} \stackrel{?}{=} T_{C_1} \cdot C_1^{c_p}$$
$$pk_T^{z_r} \cdot g_1^{z_m} \stackrel{?}{=} T_{C_2} \cdot C_2^{c_p}$$
$$\mathsf{H}_{\mathbb{G}_1}(\texttt{"rev"} \parallel w)^{z_m} \stackrel{?}{=} T_R \cdot R_w^{c_p}$$

If the checks pass, the verifier is assured that $E = (C_1, C_2)$ encrypts $g_1^{m_{\mathsf{id}}}$ and that R_w is correctly formed using the same m_{id} certified by the EC.

B Stateless Verification Procedure

This appendix specifies a concrete stateless verifier procedure for V-PASS for packets that carry WObj_w. The procedure treats the service context ctx as application-derived metadata (e.g., from a message type or service identifier) and assumes a standard freshness policy for window indices (Section 4.1). The NIZK verification subroutine invoked by the verifier is specified in Appendix A.

(1) **Input:** payload P, window index w, signature σ_P, window object $\mathsf{WObj}_w = (\mathsf{nym}_w, \sigma_{\mathsf{SE}}, E = (C_1, C_2), R_w, \pi)$, public parameters pp, verification key gpk, blocklist $\mathsf{Blocklist}_w$.
(2) **Output:** accept or reject.
(3) Derive $\mathsf{ctx} \leftarrow \mathsf{DeriveCtx}(P)$.
(4) Reject if $\mathsf{Fresh}(w) = 0$ under the deployment's timestamp/window policy.
(5) Reject if $\mathsf{ASE.Verify}(gpk, \texttt{rate} \parallel w \parallel \mathsf{ctx} \parallel \mathsf{nym}_w, \sigma_{\mathsf{SE}}) = 0$.
(6) Compute $\mu \leftarrow \mathsf{H}_{\mathsf{msg}}(\texttt{msg}, w, \mathsf{ctx}, \mathsf{nym}_w, P)$.
(7) Reject if $\mathsf{Sig.Verify}(\mathsf{nym}_w, \mu, \sigma_P) = 0$.
(8) Reject if $R_w \in \mathsf{Blocklist}_w$.
(9) Reject if $\mathsf{NIZK.Verify}(\mathsf{pp}, \mathsf{ctx}, w, \mathsf{nym}_w, \sigma_{\mathsf{SE}}, C_1, C_2, R_w, \pi) = 0$.
(10) **Accept.**

Algorithm B.1: Stateless verification for one received packet.

C Canonical Transcript Encoding and Domain Separation

The Fiat–Shamir challenge in V-PASS is computed as $c \leftarrow \mathsf{H}_{FS}(\mathsf{tr}_w)$ over the ordered transcript in Eq. 5. This section defines a canonical, unambiguous encoding for all fields in tr_w to ensure that honest provers and verifiers reconstruct identical byte strings.

Encoding conventions. Let $\mathsf{enc}(\cdot)$ be a deterministic encoder. We write $\mathsf{I2OSP}(x, \ell)$ and $\mathsf{OS2IP}(\cdot)$ for standard integer–octet conversions, and fix $\ell_w = 8$ (window index width) and $\ell_{\mathsf{len}} = 4$ (length-prefix width). Let $\ell_p = \lceil \log_2 p/8 \rceil$ and $\ell_{qse} = \lceil \log_2 q_{se}/8 \rceil$.

- **Window index:** $\mathsf{enc}(w) = \mathsf{I2OSP}(w, \ell_w)$.
- **Scalars:** for $x \in \mathbb{Z}_p$, $\mathsf{enc}(x) = \mathsf{I2OSP}(x, \ell_p)$; for $x \in \mathbb{Z}_{q_{se}}$, $\mathsf{enc}(x) = \mathsf{I2OSP}(x, \ell_{qse})$.
- **Bitstrings/strings:** $\mathsf{enc}(b) = \mathsf{I2OSP}(|b|, \ell_{\mathsf{len}}) \parallel b$, where $|b|$ is the byte length. (Strings such as ctx are first encoded as UTF-8 byte strings.)
- **Group elements:** Curves' compressed representation (e.g., BLS12-381 $\mathbb{G}_1$, SECP256R1).
- **Tuples:** concatenation with length-prefixes, as in Eq. 5.

Implementations may use any wire format as long as the verifier reconstructs the exact same byte string $\mathsf{enc}(\mathsf{tr}_w)$ for H_{FS}.

Domain separation. All hashes in pp use explicit tags (Section 6). For Fiat–Shamir, we recommend computing $\mathsf{H}_{FS}(\mathsf{tr}_w) := \mathsf{H}(\texttt{fs} \parallel \mathsf{enc}(\mathsf{tr}_w))$, for a hash H modeled as a random oracle in our analysis, and defining $c_p := \mathsf{OS2IP}(c) \bmod p$ as in Appendix A.

D Expanded Privacy Argument

This appendix expands the privacy reasoning by decomposing transcript unlinkability into independent cryptographic components. We consider a PPT adversary that observes accepted transcripts and does *not* possess TA decryption capability. As in Section 3.2, we only claim privacy for honest (non-compromised) vehicles.

Lemma D.1 (Ciphertext privacy). *For a fixed honest vehicle, the trace ciphertext $E = (C_1, C_2)$ is computationally indistinguishable from an encryption of a random group element to any non-TA observer, under DDH in $\mathbb{G}_1$.*

Proof sketch. By construction, E is ElGamal encryption in $\mathbb{G}_1$ with fresh randomness $r \leftarrow \mathbb{Z}_p$ each window. IND-CPA security follows from DDH, so E reveals no efficiently testable information about m_{id} to a non-TA observer. □

Lemma D.2 (Cross-window unlinkability of revocation tags). *Let $R_w = \mathsf{H}_{\mathbb{G}_1}(\texttt{rev} \parallel w)^{m_{\mathsf{id}}}$. In the random-oracle model for $\mathsf{H}_{\mathbb{G}_1}$ and under DDH in $\mathbb{G}_1$, the joint distribution of $\{R_w\}_w$ for an honest vehicle is computationally unlinkable across distinct windows to any party without m_{id}.*

Proof sketch. Each window uses a distinct base $\mathsf{H}_{\mathbb{G}_1}(\texttt{rev} \parallel w)$ modeled as a random group element. Without m_{id}, checking if two tags share an exponent reduces to a DDH-style relation in $\mathbb{G}_1$ under random bases. □

Lemma D.3 (Transcript privacy under ZK and anonymous attestation). *Assume: (i) the composed proof π is zero-knowledge in the random-oracle model, and (ii) the SE attestation primitive (ASE) is unlinkable and does not reveal a stable per-device identifier across signatures under gpk. Then, for honest vehicles, transcripts from distinct windows are computationally unlinkable to any non-TA observer beyond public side information (w, ctx).*

Proof sketch. By zero-knowledge, π reveals no witness information beyond the statement inputs (which already include E and R_w). By the two lemmas above, (E, R_w) reveal no efficiently testable identifier for m_{id} across windows to a non-TA observer. Finally, ASE unlinkability prevents σ_{SE} from acting as a stable device identifier. Therefore, any efficient cross-window linkage strategy would contradict at least one of these assumptions. □