

Article

A Lightweight Post-Quantum Anonymous Attestation Framework for Traceable and Comprehensive Privacy Preservation in VANETs

Esti Rahmawati Agustina ¹, Kalamullah Ramli ¹, Ruki Harwahyu ¹, Teddy Surya Gunawan ²,
Muhammad Salman ^{1,*}, Andriani Adi Lestari ¹ and Arif Rahman Hakim ³

¹ Department of Electrical Engineering, Universitas Indonesia, Depok 16424, Jawa Barat, Indonesia; esti.rahmawati@ui.ac.id (E.R.A.); kalamullah.ramli@ui.ac.id (K.R.); ruki.h@ui.ac.id (R.H.); andriani.adi@ui.ac.id (A.A.L.)

² Department of Electrical and Computer Engineering, Kulliyyah of Engineering, International Islamic University Malaysia, Kuala Lumpur 50728, Malaysia; tsgunawan@iiu.edu.my

³ Department of Cybersecurity, Politeknik Siber dan Sandi Negara, Bogor 16120, Jawa Barat, Indonesia; arif.hakim@poltekssn.ac.id

* Correspondence: muhammad.salman@ui.ac.id

Abstract

Vehicular ad hoc networks (VANETs) require authentication systems that balance privacy, scalability, and post-quantum security. While lattice-based V-LDAA offers quantum resistance, it faces challenges in signature size, traceability, and integration. We propose post-quantum traceable direct anonymous attestation (PQ-TDAA), combining National Institute of Standards and Technology (NIST)-standard Dilithium2 and Falcon-512 signatures with adapted Beullens-style blind signatures and Fiat–Shamir simplified Schnorr proofs, reducing proof size by 69.2% (8 kB vs. V-LDAA's 26 kB) and supporting European Telecommunications Standards Institute Technical Specification (ETSI TS) 102 941-compliant traceability through Road Side Unit (RSU)-assisted verification. Evaluated using SageMath, Python 3.11, and NS-3, PQ-TDAA-Falcon-512 achieves 8.1 ms and 49.7 ms end-to-end delays at 10 and 20 vehicles, respectively, with 64.7 Mbps goodput on congested 802.11p channels, showing promise for densities of ≤ 50 vehicles and advantages over Dilithium2. Real-world validation on ARM Cortex-A76 (Raspberry Pi 5, emulating automotive OBUs) yields sub-0.5 ms V2V cycles within 100 ms beacon intervals, supporting practical embedded deployment. Future work will extend PQ-TDAA to emerging 5G and NR-V2X settings, integrate more realistic mobility and channel models through coupled NS-3 and SUMO co-simulation, and investigate side-channel resistance for enhanced scalability and robustness in real deployments.

Keywords: vehicular ad hoc networks (VANETs); post-quantum cryptography (PQC); lattice-based cryptography; anonymous attestation; privacy-preserving authentication; traceability



Academic Editor: Shay Gueron

Received: 1 December 2025

Revised: 29 January 2026

Accepted: 17 February 2026

Published: 2 March 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Vehicular ad hoc networks (VANETs) are a foundational enabler of intelligent transportation systems (ITSs), enabling crucial vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication, with applications ranging from safety alerts to traffic coordination [1]. However, due to their open and dynamic nature, characterized by high mobility and topology changes, they are vulnerable to eavesdropping, impersonation,

and denial-of-service attacks [2]. In addition to security, privacy is an important issue, as continuous location data transmission can expose drivers to tracking threats [3]. Achieving this balance between privacy protection and regulatory accountability constitutes a fundamental challenge in contemporary VANET architecture.

Authentication is essential at both the node and message levels in VANET [4]. The early authentication mechanisms were based on traditional public key infrastructure (PKI)-based certificates [5], but fixed identities enabled tracking and compromised privacy. To address this issue, pseudonym-based solutions were developed; however, managing large numbers of short-term certificates and revocations also introduced additional complexity. To overcome these constraints, Brickell et al. [6] proposed cryptographic schemes that associate authentication with anonymity to achieve anonymous authentication, such as direct anonymous attestation (DAA).

The development of quantum computing creates a more basic threat. It makes modern cryptographic assumptions, such as those that support current VANET security, vulnerable to polynomial-time attacks using Shor's and Grover's algorithms [7]. This condition enables the use of post-quantum cryptography, including lattice-based designs, owing to their efficiency and security guarantees. Recent lattice-based schemes like V-LDAA [8] satisfy post-quantum security requirements. However, operational deployment has revealed critical gaps, including cryptographic payload limitations that constrain network scalability, a lack of ETSI TS 102 941-compliant conditional traceability for identifying misbehavior while preserving privacy, and an architectural rigidity that ignores distributed infrastructure capabilities. To formalize these challenges, we identified two primary barriers: (1) a loss of scalability due to the trade-off between post-quantum security and real-time processing requirements, and (2) a lack of conditional accountability as mandated by the ETSI TS 102 941 standard. These gaps motivated the following research questions: Can a post-quantum DAA scheme achieve sufficient computational efficiency to support high-frequency beaconing in dense traffic? Can such a mechanism provide a robust revocation and traceability framework without undermining user unlinkability during normal operations?

To address these questions, we propose post-quantum traceable direct anonymous attestation (PQ-TDAA). This modular lattice-based authentication framework systematically integrates established post-quantum primitives to satisfy VANET-specific requirements. While these post-quantum primitives are individually well studied, their composition for vehicular anonymous attestation poses nontrivial design challenges that, to the best of our knowledge, have not been comprehensively addressed in the literature. Our framework integrates three principal design elements. First, we adapted the simplified Schnorr protocol enhanced with the Fiat–Shamir transform [9,10], drawing upon rejection sampling techniques from Lyubashevsky's framework [11,12] within Beullens et al.'s blind signature construction [13], achieving substantial signature size reduction (from 22 KB to 8 KB). Second, we implemented threshold-based conditional traceability that satisfies the requirements of ETSI TS 102 941 [14], thereby enabling regulatory-compliant accountability without compromising user privacy in routine operations. Third, we introduced roadside unit (RSU)-assisted hierarchical verification, distributing authentication workload across the network infrastructure to reduce the computational burden on on-board units (OBUs). PQ-TDAA adopts National Institute of Standards and Technology (NIST)-standardized post-quantum cryptography (PQC) algorithms: the module lattice key encapsulation mechanism (ML-KEM)-512/advanced encryption standard (AES)-256-Galois/Counter Mode (GCM) hybrid encryption for secure channel establishment and CRYSTALS-Dilithium2 and Falcon-512 digital signatures optimized for VANET speed–size tradeoffs.

The main contributions of this work are as follows:

- A systematic integration framework for post-quantum traceable DAA in VANETs (PQ-TDAA) that integrates NIST-standard ML-KEM-512, Dilithium2, and Falcon-512 with Beullens et al.'s two-round blind signature transformed through a Fiat–Shamir-based simplified Schnorr proof is established. PQ-TDAA achieves practical efficiency by optimizing parameters for vehicular conditions, enabling complete pseudonym lifecycle management, reducing communication overhead, and controlling traceability.
- A lightweight blind signature for VANETs is obtained. We adapt Beullens et al.'s [13] lattice-based blind signature by replacing the original Lyubashevsky, Nguyen, and Plancon (LNP) zero-knowledge protocol (ZKP) with a Fiat–Shamir-transformed simplified Schnorr proof. This systematic adaptation reduces the proof size to 8 kB, which is a 63.6% reduction from Beullens et al.'s [13] original scheme and 69.2% smaller than the state-of-the-art V-LDAA [8], making credential issuance practical for bandwidth-constrained control channels.
- A complete pseudonym lifecycle with ETSI TS 102 941 compliance is obtained [14]. We introduce the first post-quantum framework covering all five operational phases (Issuance, Usage, Changing, Resolution, and Revocation) that aligns with ETSI TS 102 941 standards [14]. By integrating RSU-assisted hierarchical verification, PQ-TDAA shifts the computational burden from OBUs to the infrastructure.
- Rigorous hybrid security verification is achieved. We conducted a comprehensive three-layer validation that combines Scyther-based formal verification, informal security justification, and lattice-level cryptographic analysis. The Scyther verification under the Dolev–Yao model confirms secrecy, authentication, and synchronization with no attacks detected, while the informal analysis validates eight security and privacy properties across protocol phases. Complementarily, the lattice-based proof establishes soundness, zero-knowledge, and unforgeability for Module-Short Integer Solution (M-SIS)/Module-Learning with Errors (M-LWE), supported by a BKZ-728 parameterization providing approximately 125-bit quantum security, substantially exceeding NIST Level-1 requirements.
- Real-world deployment validation on automotive-grade ARM hardware is performed. We present direct measurements on an ARM Cortex-A76 (Raspberry Pi 5) platform, representing realistic automotive OBU platforms. Both Dilithium2 and Falcon-512 variants achieve sub-0.5 ms complete V2V cycles (beacon generation and verification), well within the 100 ms beacon interval, confirming real-time suitability for vehicular deployment. These measurements bridge the gap between theoretical design and practical embedded implementation.
- Holistic performance validation is completed. Analytical profiling and NS-3 simulations demonstrate significant gains in computational efficiency, communication compactness, and network goodput. The results reveal that signature compactness, rather than cryptographic speed, is the dominant performance factor in post-quantum vehicular networks, effectively bridging the gap between theoretical security and practical implementation.

The remainder of this paper is organized as follows. Section 2 reviews related work, covering NIST PQC algorithms, lattice-based blind signatures, hybrid post-quantum encapsulation, and post-quantum privacy-preserving authentication in VANET. Section 3 presents the PQ-TDAA framework, detailing its protocol phases from setup to revocation, along with the rationale for its design. Section 4 provides security analysis through formal verification and informal justification, as well as lattice-based security analysis. Section 5 evaluates performance with respect to computational cost, communication overhead, the real-time feasibility of V2V operations in the OBU, and network performance. Section 6 concludes the paper.

2. Related Work

2.1. National Institute of Standards and Technology (NIST) Post-Quantum Cryptography (PQC) Algorithms Standard [15,16]

This subsection describes the NIST PQC Algorithms Standard [15,16], i.e., ML-KEM-512, Dilithium2, and Falcon-512 (draft). Table 1 shows a comprehensive comparison of the NIST PQC Algorithms Standard's cryptographic parameters.

Table 1. Cryptographic Parameters of PQC Algorithms Standard [15,16].

Parameter	ML-KEM-512	Dilithium2	Falcon-512
NIST security level	1	2	1
Quantum security (bits)	143	140	141
Classical security (bits)	153	151	154
Lattice dimension (n)	256	256	512
Modulus (q)	3329	8,380,417	12,289
Public key size (bytes)	800	1312	897
Private key sizes (bytes)	1632	2528	1281
Ciphertext size (bytes)	768	-	-
Signature size (bytes)	-	2420	666

2.1.1. Module Lattice Key Encapsulation Mechanism (ML-KEM)-512

ML-KEM-512 (standardized as Federal Information Processing Standards (FIPS) 203) [15] is a post-quantum KEM designed to establish secure session keys for entities in the PQ-TDAA model. The security foundation rests on the learning with errors (LWE) problem.

2.1.2. CRYSTAL-Dilithium2 (Dilithium2)

Dilithium2 (standardized as FIPS 204) [16] is a lattice-based digital signature scheme that enables secure authentication and pseudonym certification in vehicular networks. The construction of Dilithium2 relies on the computational hardness of the M-SIS and M-LWE problems.

2.1.3. Falcon-512

Falcon-512 [17,18] is a lattice-based digital signature algorithm optimized for compactness and speed, making it well suited for vehicle-to-vehicle (V2V) broadcast authentication. Falcon is the NIST PQC standard selection that will be codified under FIPS 206 (draft). The scheme is based on the hardness of the NTRU lattice problem and employs efficient fast-Fourier sampling techniques to enable high-performance signing and verification.

2.1.4. Hardware Scaling Projection

PQC benchmarking is typically conducted on high-performance simulation environments (Intel Core i7 [19–25] @2.5–2.6 GHz, AMD Ryzen @3.3 GHz). In contrast, medium-to-high-end automotive OBUs, representative of next-generation V2X deployments, operate under significantly constrained conditions, typically utilizing ARM Cortex-A76 quad-core processors at 2.4 GHz [26,27]. Bridging this architectural performance gap represents a significant challenge in extrapolating simulation results to real-world deployment. To address this, we adopt the cycle-based normalization approach established by the pqm4 framework [28], which decouples algorithmic efficiency from frequency and memory disparities. This methodology relies on the canonical central processing unit (CPU) performance model, which defines execution time as [29]:

$$\text{Execution Time} = I \times \text{CPI} \times C \quad (1)$$

Equation (1) highlights that execution time is a function of instruction count (I), architectural efficiency (CPI), and clock speed (C). To accurately project performance from our high-performance simulation environment (AMD Zen 3) to the target automotive OBU (ARM Cortex-A76), we derived a total scaling factor based on microarchitectural disparities between the two platforms.

First, we address the efficiency gap. The AMD Zen 3 architecture features a wide execution engine capable of sustaining a high instruction per cycle (IPC) rate of 2.0 [30]. In contrast, the automotive Cortex-A76 utilizes a more constrained, energy-efficient 4-way superscalar design, resulting in a lower estimated IPC of 1.7 [26,27]. This fundamental difference in computational throughput is a critical variable in our projection model.

Second, we validate the reliability of this extrapolation using the pqm4 framework [28] and empirical findings from La Manna et al. [31]. While pqm4 [28] establishes the standard for cycle-based normalization, La Manna et al. [31] provided the physical ground truth by benchmarking PQC on actual automotive hardware (Xilinx Zynq UltraScale+). Their results confirmed that the performance trends between x86 desktops and ARM-based automotive chips were linearly preserved. This linearity demonstrates that applying a constant analytical scaling factor is a scientifically legitimate method for predicting OBU performance. The detailed parameters for this projection are summarized in Table 2.

Table 2. Hardware Scaling Factors for Automotive PQC Implementation.

Scaling Factor	Simulation Workstation (Baseline)	Automotive OBU (Target)	Scaling Ratio
Processor model	AMD Ryzen 9 5900HX (Zen 3)	ARM Cortex-A76 (R-Pi 5)	-
Clock frequency	3.3 GHz (base clock)	2.4 GHz	1.38×
Microarchitecture (IPC)	~2.0 (Zen 3 high-performance)	~1.7 (A76 4-way superscalar)	1.18×
Active core count	8 cores	4 cores	2×
Total scaling factor	Reference (1.0)	Projected gap	3.26×

As shown in Table 2, the primary performance disparity stems from the clock frequency (1.38×) and the physical core count (2×). However, a microarchitectural efficiency ratio of 1.18 represents a critical correction factor that is often overlooked in purely frequency-based comparisons. By aggregating these three vectors ($1.38 \times 1.18 \times 2.00$), we establish a total scaling factor of 3.26 times.

This 3.26× multiplier, applied throughout Section 5, scales Ryzen measurements to realistic automotive OBU performance. Empirical validation on a Raspberry Pi 5 with a Cortex-A76 processor confirmed projection accuracy (Section 5.6), ensuring that the feasibility analysis reflects actual hardware constraints rather than optimistic simulation assumptions.

2.2. Selected Existing Blind Signature and Zero-Knowledge Proof (ZKP) Mechanisms

Beullens et al. [13] proposed two rounds of communication for lattice-based blind signatures, which are optimal for maintaining latency in vehicular communication. However, the original design implemented the LNP ZKP mechanism [32], which was developed to demonstrate multiple statements simultaneously via amortization, with a per-statement complexity of $O(\log N)$. This is the most appropriate method for anonymous credential systems, in which a user may have multiple credentials. Nevertheless, VANET is severely constrained by LNP: (i) The overall proof size is 22 kB, which significantly exceeds the payload budget of safety beacons, and amortization brings no benefit when each vehicle uses only a single credential; and (ii) the concrete non-interactive ZKP instantiation, though non-interactive in the random-oracle model, has relatively heavy prover costs that translate

into additional latency for safety-critical V2X messages. To overcome these limitations, we replaced the LNP22-based proof component with a simplified Schnorr protocol made non-interactive via the Fiat–Shamir transform [9,10] and tailored rejection-sampling in the spirit of Lyubashevsky’s framework [11,12], reducing the proof size from 22 kB to about 8 kB ($\approx 63.64\%$ smaller)

The pseudocode (Algorithm A1, Algorithm A2, Algorithm A3, Algorithm A4 and Algorithm A5) illustrates the simplified Schnorr protocol with the Fiat–Shamir transform, as implemented in the framework presented in Appendix A. The protocol involves three roles: prover, responsible for generating the ZKP; verifier, who validates the proof; and signer, who issues the credential after successful verification, as illustrated in Figure 1.

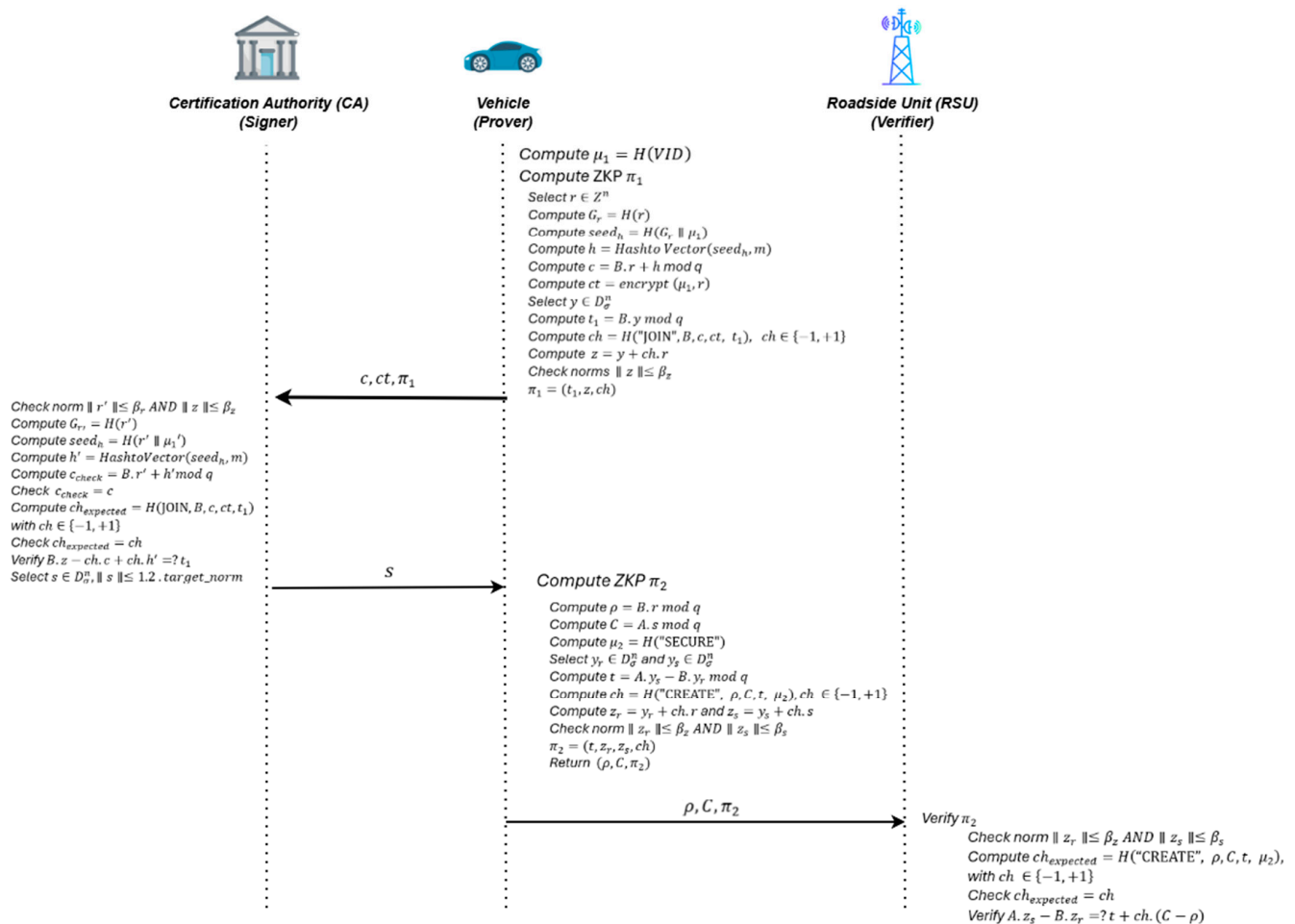


Figure 1. Simplified Schnorr blind signature protocol between CA, vehicle, and RSU in PQ-TDAA.

2.3. Hybrid Post-Quantum Encapsulation and Authenticated Data Protection

To date, NIST has not standardized a post-quantum encryption scheme; therefore, PQ-TDAA adopts an ML-KEM to derive symmetric keys and combines it with AES-256-GCM as the AEAD primitive. AES-256-GCM [33,34] is consistent with its widespread use in hybrid public key encryption and, as shown in our experiments, achieves lower computational cost than alternatives such as ChaCha20-Poly1305 and AES-256-CCM. The resulting ML-KEM–AES-256-GCM hybrid encryption is used exclusively for confidential unicast communications in PQ-TDAA, namely the Join, Network Registration, Pseudonym Changing, Resolution, and Revocation phases. Table 3 presents the comparison of total cryptographic time.

Table 3. Comparison of Total Cryptographic Time and Stability of the Hybrid Scheme.

Scheme	Cryptographic Time (Mean) (ms)	Standard Deviation (ms)	Overhead (Bytes)	Assessment
ML-KEM + AES-256-GCM	0.861	0.049	796	Optimal (fastest and most stable)
ML-KEM + Chacha20-Poly1305	1.093	0.056	796	Strong alternative (software consistent)
ML-KEM + AES-256-CCM	2.821	0.121	796	Slowest hybrid (higher variability)

Note: Execution times measured in Python 3.11 environment on Intel Core i7-8550U @ 1.80 GHz with 16 GB RAM.

2.4. Post-Quantum Privacy-Preserving Authentication in VANETs

DAA [6] emerged as a robust cryptography that achieves security and privacy through anonymous authentication. In an early implementation of DAA-based systems, Whitefield et al. [35] used a trusted component to produce unlinkable pseudonyms but had a high reliance on the integrity of a host unit. Alternatively, Chen et al. [36] suggested threshold anonymous announcement (TAA), which is a combination of DAA and threshold authentication. Although TAA improved message reliability, the scheme introduced significant communication and coordination overhead, making it less applicable in dynamic VANET settings. Desmoulins et al. [37] suggested pseudonyms based on pre-DAA, in which the pseudonym lifecycle is transferred to a secure element, so that the host dependency can be eradicated and beacon-compatible speeds (under 50 ms) can be achieved. However, despite its efficiency, the protocol proved susceptible to impersonation, replay, and key-substitution attacks. To address the quantum threat to classical assumptions, lattice-based DAA protocols were proposed, culminating in V-LDAA [8] as a lattice-based DAA scheme explicitly designed for VANETs.

Beyond the DAA family, various post-quantum and privacy-preserving authentication schemes have been explored for VANETs. Digital signature-based schemes are widely used to realize conditional privacy-preserving authentication (CPPA) [22–25,38,39], where vehicles authenticate safety messages while protecting long-term identities through pseudonyms or short-term certificates. Ring signatures [20] provide signer ambiguity within an authorized group, while traceable ring signatures [40] balance anonymity and accountability by permitting the authorized tracing of misbehaving vehicles. Identity-based signature schemes [19] achieve authentication without relying on conventional PKI, as public keys are derived from identities or pseudo-identities managed by a trusted authority. Revocable ring signatures [39] further support the efficient revocation of compromised or misbehaving nodes. Attribute-based signatures [41] enable fine-grained authentication by verifying specific vehicle or RSU attributes without exposing actual identities. Finally, blind signatures [8] guarantee that the signer does not learn the content of the message being signed, and are typically used in scenarios requiring strong privacy when the signer must not be able to link signatures to specific messages; this primitive underpins DAA and is adapted in PQ-TDAA to achieve anonymous authentication while enabling conditional traceability via trace tokens.

Within this landscape, DAA and blind-signature-based constructions are the most relevant baselines for PQ-TDAA, as they explicitly target anonymous authentication, specifically membership attestation, with a structured pseudonym lifecycle. In contrast, ring-signature, traceable-ring, identity-based, attribute-based, and revocable-ring schemes primarily provide privacy-preserving message authentication under different security abstractions. Within the broader development of post-quantum and privacy-preserving

authentication, constructions based on DAA and blind signatures are the most relevant foundations for this study, as they provide anonymous attestation supported by a structured pseudonym lifecycle rather than merely protecting message-level privacy. Nevertheless, the V-LDAA scheme exhibits three significant challenges in realistic vehicular networks: large signature and proof sizes that create bandwidth bottlenecks on the IEEE 802.11p control channel, the absence of conditional traceability compliant with ETSI TS 102 941, and a rigid architecture that does not efficiently utilize roadside infrastructure. The comparative analysis in Table 4 shows that none of the existing approaches, including classical DAA variants, pre-DAA, V-LDAA, or ring-signature-based mechanisms, can simultaneously ensure privacy, accountability, efficiency, and quantum resistance in high-frequency vehicular safety communication. These unresolved limitations define the practical gap that motivates the design of the proposed PQ-TDAA framework.

Table 4. Strengths and Limitations of Existing Schemes.

DAA Variants	Key Features	Strengths	Limitations
Classical DAA [6]	ZKP, trusted platform module-based (TPM-based)	Strong privacy and accountability	Heavy computation, less suitable for practical real-time VANET
TAA [36]	Threshold authentication with TAA	Ensures message reliability, supports auditability	Complex coordination, additional communication overhead
Privacy-enhanced DAA for VANETs [35]	Integrates DAA into VANET PKI	Strong privacy with conditional traceability	High computation overhead, scalability issues
Pre-DAA-based pseudonym [37]	Secure elements handle the entire pseudonym lifecycle	Real-time, feasible, efficient, and flexible driver vehicle mapping	Weak Join phase; weak against impersonation, replay, and quantum attacks; incomplete pseudonym lifecycle
V-LDAA [8]	Optimized lattice signatures, smaller proof size	Long-term security is more efficient than early lattice schemes	High communication overhead, challenging for VANET bandwidth/latency, incomplete pseudonym lifecycle, and lack of infrastructure integration

3. Proposed Framework Development

This section presents the design of the proposed PQ-TDAA framework for secure and privacy-preserving V2X communication in resource-constrained VANET environments. We define security and privacy requirements based on our previous work in [3,42]. PQ-TDAA operates within a hierarchical VANET architecture comprising certification authority (CA), law enforcement agency (LEA), roadside units (RSUs), and vehicles equipped with OBU and trusted platform module (TPM) 2.0 modules. The communication model assumes an IEEE 802.11p-based V2V/V2I system with a 300–500 m range, a 10 Hz beacon frequency, and <100 ms latency for safety-critical messages. For trust assumptions, CA and LEA are fully trusted; RSUs are semi-trusted (honest-but-curious); and vehicles are untrusted, with TPM providing hardware-anchored key protection.

We consider a Dolev–Yao adversary augmented with quantum capabilities—able to eavesdrop, inject, and replay messages, execute Sybil attacks, and break classical cryptographic assumptions via Shor’s algorithm, but unable to compromise CA’s master secrets or TPM-protected keys. This model ensures that evaluation reflects realistic deployment constraints while capturing post-quantum adversarial capabilities. Table 5 describes all

phases, entities, and their interactions in the PQ-TDAA framework, and Figure 2 illustrates the interactions among entities within it.

Table 5. Eight Phases, Four Entities, and Their Interactions in the Proposed Framework.

Phase	Entities	Interaction	Entities' Role
Setup	CA	Securely generate and transmit the security parameter and cryptographic keys to other entities	All entities securely store the security parameter and cryptographic keys.
Join	CA and vehicle	1. Enroll the request credential by vehicle 2. CA issues a credential	CA authenticates the OBU anonymously, generates a trace token and a vehicle internal identity, and issues a credential for a blind signature.
Create	Vehicle	3. Generate a pseudonym credential	OBU and TPM are embedded in the vehicle. TPM is responsible for all cryptographic and security operations, and OBU is responsible for V2I and V2V communication.
Network Registration (V2I communication)	Vehicle and roadside unit (RSU)	4. Network registration request 5. Accepted/rejected request	Vehicle requests to enter the vehicular network. RSU authenticates the vehicle anonymously and signs the pseudonym credential.
Broadcast Beacon (V2V communication)	Vehicles	6. Broadcast message 7. Verify signature	The vehicle creates and signs the message beacon, then broadcasts it to nearby vehicles. The receiving vehicles verify the signatures of the beacon and the pseudonym credential.
Pseudonym Changing	Vehicles and RSU	8. Pseudonym changing request 9. Issue a new pseudonym credential	Vehicle initiates pseudonym changing. RSU generates and signs a new pseudonym credential.
Pseudonym Resolution	RSU and LEA	10. Pseudonym resolution request 11. Obtained a malicious vehicle	RSU initiates pseudonym resolution upon detecting a malicious vehicle. LEA opens the trace token of a malicious vehicle.
Pseudonym Revocation	LEA and CA	12. Pseudonym revocation request	CA adds the vehicle-related identity to the blacklist server.

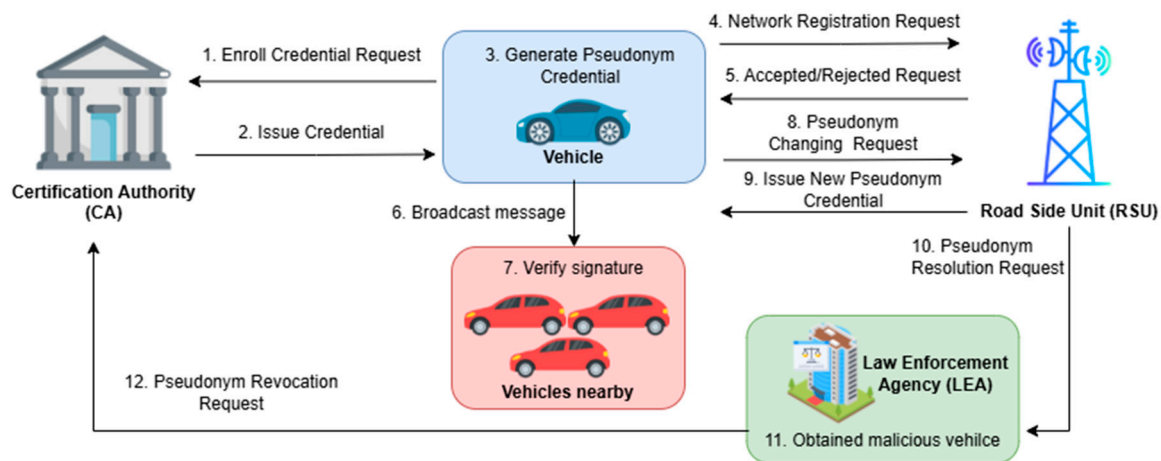


Figure 2. Our proposed PQ-TDAA framework.

3.1. Setup Phase

In the setup phase, the CA generates security parameters and a cryptographic key for all entities. The security parameter is shown in Algorithm A1 System Setup (Appendix A). The generated cryptographic key is shown in Table 6. The CA securely transmits the cryptographic keys. For the endorsement (for encryption and decryption) and digital signature key of each OBU, it is assumed to be securely pre-installed by the manufacturer during the production process, and it is secure in the TPM. The CA, RSU, and LEA store their encryption/decryption and digital signature keys securely.

Table 6. Cryptographic Keys Generated by the CA in the Setup Phase.

Cryptographic Key	Description
pk_{EKTTPM}, sk_{EKTTPM}	Public and private endorsement keys of the vehicle
pk_{CA}, sk_{CA}	Public and private key pair of the CA
pk_{RSU}, sk_{RSU}	Public and private key pair of the RSU
pk_{LEA}, sk_{LEA}	Public and private key pair of the LEA
$pk_{TPM}^{DSA}, sk_{TPM}^{DSA}$	Public and private key pair of the TPM for the digital signature mechanism
$pk_{CA}^{DSA}, sk_{CA}^{DSA}$	Public and private key pair of the CA for the digital signature mechanism
$pk_{RSU}^{DSA}, sk_{RSU}^{DSA}$	Public and private key pair of the RSU for the digital signature mechanism
$pk_{LEA}^{DSA}, sk_{LEA}^{DSA}$	Public and private key pair of the LEA for the digital signature mechanism

3.2. Join Phase

The Join phase begins with a multi-step communication process between the OBU and the CA, designed to mutually and anonymously authenticate both parties before the OBU is integrated into the vehicular network.

1. OBU generates a random nonce N_a and a hash value $\omega = H(pk_{EKTTPM})$. Then, it encrypts N_a , ω , and pk_{EKTTPM} using the CA's public key, and transmits to the CA.
2. CA decrypts the message and verifies that the included hash ω matches its own computation of $H(pk_{EKTTPM})$. Once validated, the CA generates its own random nonce N_i , derives its hash $\theta = H(N_i)$, and signs the pair (N_i, θ) with its private signature key. This response is then encrypted with the OBU's public key pk_{EKTTPM} and sent back to the OBU.
3. After decrypting the message using sk_{EKTTPM} , the OBU verifies the signature on (N_i, θ) , confirming that the message originated from the legitimate CA and establishing

a mutually authenticated channel. OBU then initiates the simplified Schnorr blind signature using the Fiat–Shamir transform. These initiation results are the random value (r), commitment (c_1), the encryption tuple (c_t), and the ZKP (π_1). The OBU encrypts c_1 , c_t , π_1 using CA's public key and the encrypted message to CA.

4. The CA decrypts the encrypted message using the CA's secret key and verifies that π_1 follows the simplified Schnorr with Fiat–Shamir transform blind signature. Upon successful verification, the CA computes a short preimage s then generates a digital signature $\vartheta = \text{sign}_{\text{CA}}(s)$ using its signing key $sk_{\text{CA}}^{\text{DSA}}$. Concurrently, the CA generates a $\text{trace}_{\text{token}}$ by encrypting a hash of a newly generated internal vehicle ID, $H(\text{VID}_{\text{in}} \parallel \text{salt})$ with the LEA's public key: $\text{trace}_{\text{token}} = \text{Enc } pk_{\text{LEA}}(H(\text{VID}_{\text{in}} \parallel \text{salt}))$. The final tuple $(s, \vartheta, \text{trace}_{\text{token}})$ is encrypted using the OBU's public key pk_{EKTPM} and sent to the OBU.
5. Upon reception, the vehicle decrypts the message to retrieve s and $\text{trace}_{\text{token}}$. It verifies the signature ϑ on s using the CA's public key $pk_{\text{CA}}^{\text{DSA}}$. If successful, the vehicle securely stores s and $\text{trace}_{\text{token}}$ for subsequent protocol operations, completing the Join phase.

3.3. Create (Pseudonym) Phase

The objective of the Create phase is for the OBU to generate the pseudonym credential. This process uses the secrets obtained during the Join phase to construct a ZKP that authenticates the OBU without revealing its real identity. The OBU then reuses s and r secrets of the Join phase to build a ZKP that allows the vehicle to demonstrate its legitimacy without revealing its true identity. The OBU generates this proof using the simplified Schnorr blind signature scheme with the Fiat–Shamir Transform (using Algorithm A4). The final output of this process is a tuple $pse_{\text{cre}} = (\rho, \pi_2, \text{trace}_{\text{token}}, t_{\text{exp}})$, where ρ is a blinding factor, π_2 is the proof, $\text{trace}_{\text{token}}$ ensures the credential is bound to the tracing authority, and t_{exp} is the credential's expiration date. This pseudonym credential is then used during the Network Registration and Broadcast Beacon phases.

3.4. Network Registration (V2I Communication) Phase

During the Network Registration phase, each OBU establishes a secure V2I connection with a nearby RSU to obtain an authenticated pseudonym.

1. The OBU produces a timestamp T_a , generates the message msg_{join} , and creates the pseudonym credential as $pse_{\text{cre}} = (\rho, \pi_2, \text{trace}_{\text{token}}, t_{\text{exp}})$. The OBU subsequently calculates a digest $pse_{\text{digest}} = H(pse_{\text{cre}})$ and encrypts pse_{digest} and pse_{cre} together with pk_{EKTPM} and T_a by the public key pk_{RSU} of the RSU.
2. On obtaining the ciphertext, the RSU decrypts it using the secret key sk_{RSU} and verifies the freshness of T_a . It recalculates $pse'_{\text{digest}} = H(pse_{\text{cre}})$ and verifies that $pse'_{\text{digest}} = pse_{\text{digest}}$. The RSU also verifies a credential's expiry by checking the t_{exp} and the validity of the proof π_2 under the FSwA mechanism. If all the checks are successful, RSU signs pse_{digest} by using its digital signature key $sk_{\text{RSU}}^{\text{DSA}}$ to produce $\text{sign}_{\text{pseu}} = \text{Sign } sk_{\text{RSU}}^{\text{DSA}}(pse_{\text{digest}})$. T_b is a new timestamp, and RSU sends the encrypted response $(T_b, pse_{\text{digest}} \parallel \text{sign}_{\text{pseu}})pk_{\text{EKTPM}}$ to the OBU.
3. The OBU retrieves T_b and the credential signature after decryption by using sk_{EKTPM} . It authenticates the RSU's signature with $pk_{\text{RSU}}^{\text{DSA}}$ by determining whether $\text{Ver } pk_{\text{RSU}}^{\text{DSA}}(pse_{\text{digest}}, \text{sign}_{\text{pseu}}) = \text{true}$ and ensuring that T_b is fresh. Upon successful verification of both verifications, the OBU accepts $\text{sign}_{\text{pseu}}$ as a valid signature and stores it for later use in beacon broadcasting.

3.5. Broadcast Beacon (V2V Communication) Phase

During the Broadcast Beacon phase, each OBU periodically broadcasts its status information to neighboring vehicles. The transmitting vehicle, referred to as OBU A, produces a timestamp T_c and forms a beacon message, which is $\text{beacon} = (\text{pos} \parallel \text{speed} \parallel \text{notif})$, where pos is the position of the vehicle, speed is the current speed of the vehicle, and notif is auxiliary safety messages. To maintain authenticity, OBU A uses its signing key sk_{TPM}^{DSA} to sign the beacon and generates $\text{sign}_{msg} = \text{Sign } sk_{TPM}^{DSA}(\text{beacon})$. Second, OBU A transmits the $(pse_{digest} \parallel \text{sign}_{pseu}, \text{sign}_{msg}, \text{beacon}, pk_{TPM}^{DSA}, T_c)$ to the passing vehicles. The sign_{pseu} and digest pseudonym credentials pse_{digest} allow the receiver to determine that the beacon is sent by an authentic vehicle, RSU-validated pseudonym, and not a fake source. When this broadcast is received by another vehicle, OBU B, the freshness of T_c is first verified to avoid replaying. It subsequently authenticates the pseudonym signature with the RSU public key by verifying that $\text{Ver } pk_{RSU}^{DSA}(pse_{digest}, \text{sign}_{pseu})$ is true. If the validation succeeds, the $\text{beacon} = (\text{pos} \parallel \text{speed} \parallel \text{notif})$ is considered a valid and reliable broadcast.

3.6. Pseudonym Changing Phase

During the Pseudonym Changing phase, the OBU periodically changes its pseudonym based on t_{exp} to avoid long-term linkage while remaining traceable via the RSU.

1. The OBU creates the message msg_{chg} and produces a timestamp T_d . It then constructs the transmission packet $((pse_{digest} \parallel \text{sign}_{pseu}), msg_{chg}, T_d, pk_{EKTTPM})$, encrypted with the RSU's public key pk_{RSU} .
2. The RSU receives the packet, decrypts it with sk_{RSU} and checks the validity of T_d . The RSU maintains a mapping between each vehicle's pseudonym credential and its digest value (pse_{digest}). From the stored credential pse_{cre} , RSU extracts the embedded tracing token $trace_{token}$ to maintain conditional traceability. RSU then uses a new credential token, $cred_{id} = \text{RAND}(128)$, and calculates another tracing token as $newtrace_{token} = trace_{token} \parallel cred_{id}$. A new pseudonym credential is then built as $newpse_{cre} = (\rho, \pi_2, newtrace_{token}, newt_{exp})$, with $newt_{exp}$ being the new expiration time. RSU calculates $newpse_{digest} = H(newpse_{cre})$, then uses its digital signature key sk_{TPM}^{DSA} to sign the new pseudonym credential digest, the result of which is $newsign_{pseu} = \text{Sign } sk_{RSU}^{DSA}(newpse_{digest})$, and creates a timestamp T_e . It then forwards encrypted data $(newpse_{digest} \parallel newsign_{pseu}, newpse_{cre}, T_e)pk_{EKTTPM}$ to the OBU.
3. The OBU receives the new pseudonym credential $newpse_{cre}$ and its signature $newsign_{pseu}$ after the decryption using sk_{EKTTPM} . It confirms that the signature of the RSU is also correct, $\text{Ver } pk_{RSU}^{DSA}(newpse_{digest}, newsign_{pseu}) = \text{true}$, and that T_e is fresh. Upon satisfying both checks, the OBU stores the signed pseudonym as its active pseudonym signature for future V2V communication.

3.7. Pseudonym Resolution Phase

The Pseudonym Resolution phase enables the authorized recovery of a digest of the vehicle's internal identity in the event of a legal investigation or a report of misbehavior. RSU is the starting point in the process and has a valid pseudonym credential pse_{cre} and its signature, sign_{pseu} .

1. The RSU creates a timestamp T_f to eliminate replay and form a resolution request message, denoted as msg_{res} . The whole package $(T_f, msg_{res}, pse_{cre}, (pse_{digest} \parallel \text{sign}_{pseu}))$ is then encrypted using the public key of LEA and sent to LEA.
2. LEA decrypts the message using its private key sk_{LEA} and the freshness of T_f is checked. Then, it verifies the authentication of the pseudonym data by verifying sign_{pseu} with the public verification key pk_{RSU}^{DSA} of the RSU to ensure that pse_{digest} is

signed correctly. After verification, LEA derives the pse_{cre} that contains the commitment ρ , the proof π_2 , the traceable token $trace_{token}$, and the expiration time t_{exp} . The LEA authenticates the validity of t_{exp} by ensuring that the credential was valid at the time of the request. It then uses its private key sk_{LEA} to decrypt the trace token, thereby retrieving the hashed vehicle internal identity $H(VID_{in})$.

3.8. Pseudonym Revocation Phase

This step is implemented when a vehicle is identified as malicious or non-compliant. After the LEA decides that a particular pseudonym must be revoked, it first creates a timestamp T_g to ensure the message's freshness. LEA forms a revocation message: $msg_{rev} = (H(VID_{in}), \text{"reason"})$. To ensure authenticity, LEA calculates the signature $sign_{rev} = \text{sign}(msg_{rev})$ by using its secret key sk_{LEA}^{DSA} . All the data $(msg_{rev} || sign_{rev}, T_g)$ are then encrypted with the public key of the CA. Upon receiving the encrypted data, the CA decrypts the message using its own key, sk_{CA} , and verifies the freshness of T_g . To verify the message received by the CA, the $sign_{rev}$ is authenticated with the pk_{LEA}^{DSA} , ensuring that the revocation request genuinely originated from the LEA. Once confirmed, the CA extracts msg_{rev} and retrieves the value of $H(VID_{in})$. Finally, this hashed vehicle identifier is appended to the server-side blacklist to prevent the revoked pseudonym from further participation in vehicular communications. CA notifies RSU of the revoked pseudonym, and RSU informs the network. The message transmission and reception among the entities in these phases are illustrated in Figure 3.

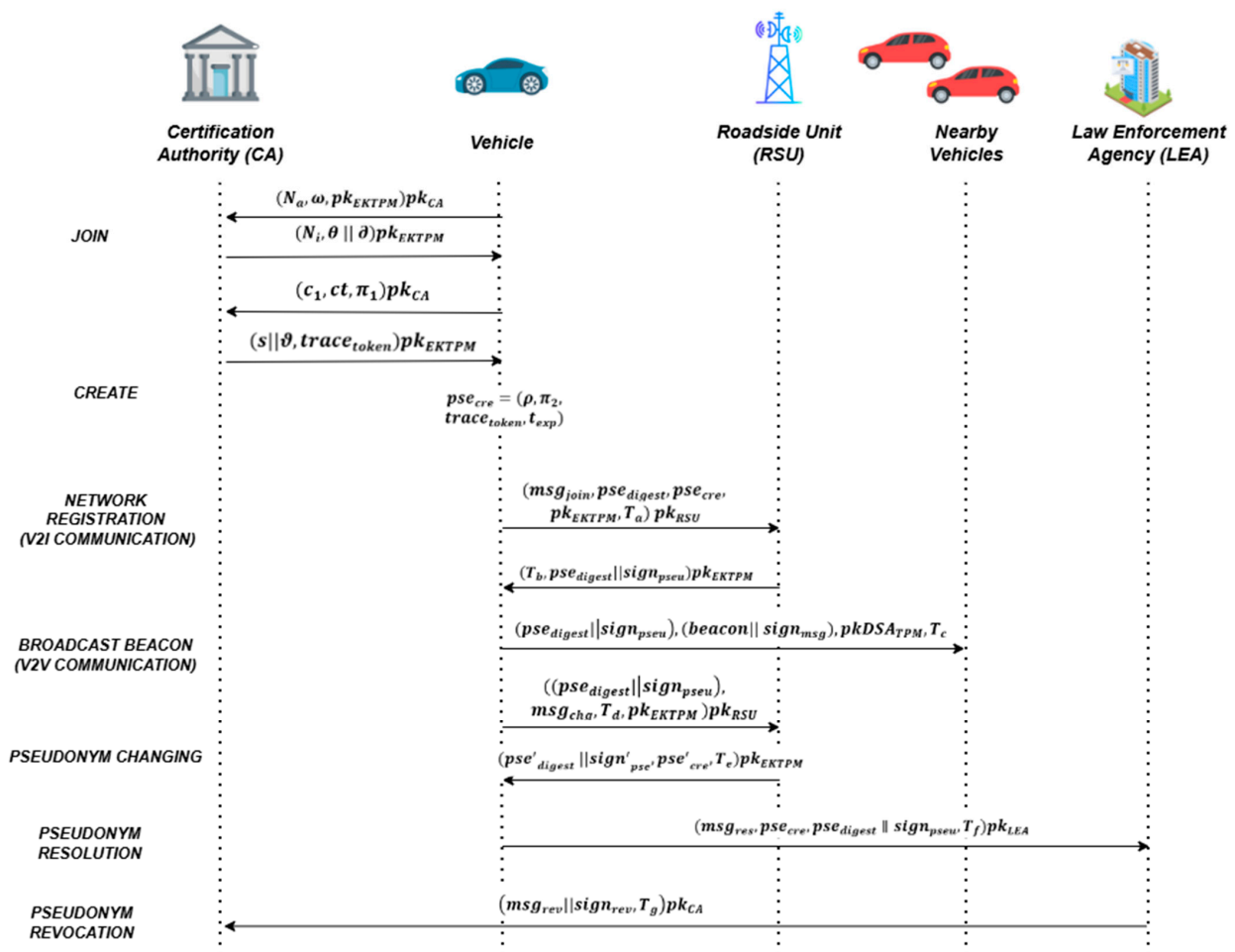


Figure 3. Message interaction between entities within the phases of our proposed PQ-TDAA framework.

3.9. PQ-TDAA Design Rationale

3.9.1. Choice of Post-Quantum Cryptography Standard Algorithms

The selection of the underlying post-quantum primitives in PQ-TDAA is driven by the need to provide at least 128-bit quantum-resistant security while remaining compatible with the stringent latency and bandwidth constraints of VANET deployments. ML-KEM-512 is adopted as the key-encapsulation component because it adheres to the NIST FIPS 203 draft profile for Level-1 security and provides a compact public key and ciphertext suitable for frequent session-key establishment between vehicles and infrastructure. Since NIST has not yet standardized a post-quantum public-key encryption scheme for data encapsulation, PQ-TDAA employs a hybrid AEAD design in which ML-KEM-512 is combined with AES-256-GCM. The experimental results in Table 2 show that this ML-KEM-512 and AES-256-GCM instantiation achieves the lowest cryptographic computation time among the evaluated hybrid options, making it particularly suitable for latency-sensitive VANET scenarios.

For digital signatures, CRYSTALS-Dilithium2 and Falcon-512 are chosen as representative NIST PQC standards that achieve similar security levels but exhibit contrasting speed–size tradeoffs: Dilithium2 provides relatively fast signing and verification with larger signatures, whereas Falcon-512 offers significantly more compact signatures at the cost of somewhat higher computational overhead. This pair allows PQ-TDAA to isolate and study the impact of signature compactness versus cryptographic speed on end-to-end VANET performance.

3.9.2. Simplified Schnorr with Fiat–Shamir Transform in PQ-TDAA

PQ-TDAA uses a simplified Schnorr proof with parameters that meet both NIST post-quantum security standards and VANET performance requirements. Instead of using the Gentry, Peikert, and Vaikuntanathan (GPV) trapdoor-based approach from the original Beullens construction, we adopted the Lyubashevsky framework [12]. This is the same cryptographic approach underlying CRYSTALS-Dilithium, but adapted for blind signature operations. This choice offers a more straightforward implementation that avoids trapdoor generation, which is critical for automotive hardware constraints.

We maintain the core parameters from Beullens et al. [13], specifically $q = 7933$ and $\sigma = 232$, but adjust the security bounds slightly higher. The bound parameters β_s and β_z are increased by $1.2\times$ to approximately 8909, which ensures robust security margins. This results in approximately 125 bits of quantum security, meeting NIST Level-1 requirements. The practical benefit is significant; our signature requires only 8 kB, representing a 63.6% reduction compared to Beullens' 22 kB. Even including the 2 kB commitment that must be sent separately, the total communication is 10 kB, which is 54.5% smaller than the original Beullens construction. Section 4.4 provides the complete security analysis. Table 7 compares our VANET-specific parameter adaptations with the original Beullens construction.

Table 7. Parameter Comparison Between Original Beullens et al. [13] and PQ-TDAA Blind Signature.

Parameter	Beullens et al. [13]	Our PQ-TDAA
Modulus q	7933	7933
Gaussian σ	232.0	232.0
β_r (uniform)	$2\sqrt{2d} \approx 64$	$2\sqrt{n} \approx 64$
β_z, β_s (Gaussian)	$\sigma\sqrt{2d} \approx 7424$	$1.2\sigma\sqrt{n} \approx 8909$
Trapdoor mechanism	NTRU + GPV sampling	None (random A, B) (Lyubashevsky style)
Credential issuance	$\tilde{s} \leftarrow A^{-1} \cdot \sigma(c)$ via trapdoor	$s \leftarrow D_\sigma(\text{random Gaussian})$
ZKP mechanism	LNP	Simplified Schnorr with Fiat–Shamir transform
Signing paradigm	GPV hash-and-sign	Fiat–Shamir (commitment-challenge-response)
Signature	$(\rho, \pi_2) : \approx 22\text{kB}$	$(\rho, \pi_2) : \approx 8\text{kB}$
Auxiliary data	-	C (2 kB)

3.9.3. Separation of Control and Data Plane

PQ-TDAA explicitly separates the control plane, where identities and credentials are established, from the data plane, where high-frequency safety beacons are disseminated. In the control plane, the lattice-based blind signature and simplified Schnorr proofs are used only in the Join, Create, and Network Registration phases to provide anonymous yet accountable authentication between the vehicle and the CA (π_1) and between the vehicle and the RSU (π_2). During these infrequent V2I interactions, the OBU demonstrates possession of valid secrets and obtains a pseudonym credential bound to a trace token, which remains unlinkable in regular operation.

Once the RSU has verified the proof and signed the pseudonym digest, subsequent V2V communications operate purely in the data plane and do not carry the heavy ZKP. Each broadcast beacon only includes the pseudonym digest pse_{digest} and its RSU signature $sign_{pseu}$, the beacon payload (pos||speed||notif), the vehicle's digital signature public key, signature of beacon, and a timestamp. This design keeps the size of safety beacons at 6254 bytes for the Dilithium2 instantiation and 2331 bytes for Falcon-512, thereby significantly reducing fragmentation and channel occupancy compared with embedding complete proofs in every beacon. By concentrating complex cryptographic operations in the control plane and keeping the data plane lightweight, PQ-TDAA can provide strong anonymous authentication without violating the 100 ms beaconing interval required for real-time VANET safety applications.

3.9.4. ETSI TS 102 941 Trust and Privacy Compliance Mapping

To demonstrate the industrial relevance of the proposed PQ-TDAA framework, this subsection maps the core trust and privacy requirements defined in ETSI TS 102 941 V2.2.1 [14] to the corresponding mechanisms implemented in our design. In our framework, the ETSI enrollment authority (EA) role is instantiated by a CA, and the authorization authority (AA) role is instantiated by a LEA that operates the authorization and pseudonym-resolution functions under appropriate legal and operational constraints. Table 8 summarizes how PQ-TDAA implements pseudonymity, conditional traceability, separation of duties between the EA and AA, and secure credential lifecycle management.

Table 8. ETSI TS 102 941 Trust and Privacy Compliance of the Proposed PQ-TDAA Framework.

ETSI Requirement	Clause	PQ-TDAA Mechanism
Pseudonymity for ITS-Station (ITS-S) with privacy; AA must not re-identify vehicles (ItssWithPrivacy).	3.4 Notation: ItssWithPrivacy (“pseudonymity has to be assured and re-identification by the AA is not allowed”).	PQ-TDAA issues lattice-based pseudonym certificates via blind signatures, ensuring the CA never learns or reconstructs the privacy-enabled vehicle's canonical identifier.
Privacy based on pseudonymity and unlinkability; the canonical identifier is never transmitted over the air.	5 Privacy in ITS: pseudonymity “but can still be accountable for that use”; unlinkability; “never transmitting the station's canonical identifier in communications between ITS stations.”	In the Join phase, the CA issues a secret for local pseudonym derivation to keep the vehicle's real identity private. The RSU then registers this pseudonym without identifying the vehicle, allowing it to sign safety messages using NIST post-quantum signatures (Falcon/Dilithium). Finally, neighboring vehicles use the RSU's public key to verify these messages, ensuring secure authentication while maintaining total anonymity.

Table 8. Cont.

ETSI Requirement	Clause	PQ-TDAA Mechanism
Separation of duties between EA (identity) and AA (authorization) to protect privacy.	5 Privacy in ITS: privacy of registration ensured by limiting canonical identifier knowledge to EAs and by separating EA and AA roles.	PQ-TDAA strictly separates the CA, which performs anonymous authentication of vehicles and maintains only pseudonymous registration records without storing any long-term real-world identifiers, from the LEA, which merely processes blinded authorization requests and never gains access to either canonical identifiers or the CA's internal identity records.
Conditional accountability and traceability via a canonical identifier and a canonical key stored at EA.	5 Privacy in ITS (accountability); 6.1.2 Manufacture: EA stores canonical identifier, profile, and canonical public key as an associated set.	ETSI TS 102 941 requires EAs to store canonical identifiers, profile data, and public keys for accountability. PQ-TDAA implements this using pseudonymous registration records instead of real-world identifiers. During resolution, LEAs use trace tokens to link misbehaving pseudonyms to CA registration data, ensuring ETSI-compliant conditional traceability without long-term real-world identifier storage.
Revocation and end-of-life handling through CTL/CRL management and refusal of further credentials.	6.1.5 Maintenance; 6.1.6 End of life; 6.3 Generation, distribution, and use of CTL and CRL.	For revocation, the CA inserts the temporary identity of a misbehaving vehicle into a blacklist service and announces the corresponding pseudonym credentials to the RSUs, which then disseminate this revocation information across the network so that other vehicles can reject any subsequent beacons or protocol messages originating from the revoked pseudonym.

4. Security Evaluation

Both formal and informal approaches were used to assess PQ-TDAA's security in depth. The formal analysis was implemented using the Scyther tool [43] for the interactive phase of PQ-TDAA. Informally, the framework was demonstrated to meet important security and privacy requirements of VANETs. We also performed a security analysis on the modified version of Beullens' lattice-based blind signature.

4.1. Formal Analysis

We formally modeled and verified all interactive phases of the proposed PQ-TDAA framework—Join, Network Registration (V2I), Broadcast Beacon (V2V), Pseudonym Resolution, and Pseudonym Revocation—using the Scyther tool [43] under the standard Dolev–Yao adversarial model. Each Scyther specification provides the attacker with complete network control, perfect cryptography, unbounded session interleaving, and arbitrary message replay. The only phase not modeled is the local Create, since it is executed solely by the OBU without interaction and thus does not expose an observable protocol surface to the adversary. For every analyzed phase, all secrecy and authentication claims are satisfied, and no attacks are discovered within Scyther's exploration bounds, showing that session data and exchanged parameters remain confidential. At the same time, non-injective synchronization and agreement properties provide the required authentication between protocol entities (OBU, RSU, LEA, and CA).

The Scyther results for the PQ-TDAA Join scheme are shown in Figure 4; the Network Registration phase in Figure 5; and Pseudonym Changing, Resolution, and Revocation in Figure 6.

Claim	Status	Comments
PTQDAAJOIN T PTQDAAJOIN.T1 Secret ni	Ok	No attacks within bounds.
PTQDAAJOIN.T2 Secret m4	Ok	No attacks within bounds.
PTQDAAJOIN.T3 Secret phi2	Ok	No attacks within bounds.
PTQDAAJOIN.T4 Secret tt	Ok	No attacks within bounds.
PTQDAAJOIN.T5 Secret s	Ok	No attacks within bounds.
I PTQDAAJOIN.I1 Secret pk(T)	Ok	No attacks within bounds.
PTQDAAJOIN.I2 Secret Cs	Ok	No attacks within bounds.
PTQDAAJOIN.I3 Secret ct	Ok	No attacks within bounds.
PTQDAAJOIN.I4 Secret phi1	Ok	No attacks within bounds.

Claim	Status	Comments
PTQDAAJOIN T PTQDAAJOIN.T1 Alive	Ok	No attacks within bounds.
PTQDAAJOIN.T2 Nisynch	Ok	No attacks within bounds.
PTQDAAJOIN.T3 Niagree	Ok	No attacks within bounds.
I PTQDAAJOIN.I1 Alive	Ok	No attacks within bounds.
PTQDAAJOIN.I2 Nisynch	Ok	No attacks within bounds.
PTQDAAJOIN.I3 Niagree	Ok	No attacks within bounds.

Figure 4. Verification results of the PT-QDAA Join protocol using the Scyther tool version 1.1.3, showing all security claims marked Ok, with no attacks found within bounds.

Claim	Status	Comments
PQTDAAVNetReg O PQTDAAVNetReg.O1 Secret psig	Ok	No attacks within bounds.
PQTDAAVNetReg.O2 Secret rpsig	Ok	No attacks within bounds.
PQTDAAVNetReg.O3 Alive	Ok	No attacks within bounds.
PQTDAAVNetReg.O4 Nisynch	Ok	No attacks within bounds.
PQTDAAVNetReg.O5 Niagree	Ok	No attacks within bounds.
R PQTDAAVNetReg.R1 Secret auxc	Ok	Verified No attacks.
PQTDAAVNetReg.R2 Secret psig	Ok	Verified No attacks.
PQTDAAVNetReg.R3 Secret m1	Ok	Verified No attacks.
PQTDAAVNetReg.R4 Secret pss	Ok	Verified No attacks.
PQTDAAVNetReg.R5 Alive	Ok	Verified No attacks.
PQTDAAVNetReg.R6 Nisynch	Ok	Verified No attacks.
PQTDAAVNetReg.R7 Niagree	Ok	Verified No attacks.

Figure 5. Verification results of the PQ-TDAA Network Registration protocol using the Scyther tool version 1.1.3, showing all security claims marked Ok, with no attacks found within bounds.

Claim	Status	Comments
PQTDAApseudoChange V PQTDAApseudoChange.V1 Secret newpsdig	Ok	No attacks within bounds.
PQTDAApseudoChange.V2 Secret newpsacre	Ok	No attacks within bounds.
PQTDAApseudoChange.V3 Alive	Ok	No attacks within bounds.
RSU PQTDAApseudoChange.RSU1 Secret newpsdig	Ok	No attacks within bounds.
PQTDAApseudoChange.RSU2 Secret newpsacre	Ok	No attacks within bounds.
PQTDAApseudoChange.RSU3 Alive	Ok	No attacks within bounds.
PQTDAApseudoChange.RSU4 Nisynch	Ok	No attacks within bounds.
PQTDAApseudoChange.RSU5 Niagree	Ok	No attacks within bounds.

Claim	Status	Comments
RSU PQTDAARevoke.RSU1 Secret k(RSU,LEA)	Ok	No attacks within bounds.
LEA PQTDAARevoke.LEA1 Alive	Ok	No attacks within bounds.
PQTDAARevoke.LEA2 Niagree	Ok	No attacks within bounds.
PQTDAARevoke.LEA3 Nisynch	Ok	No attacks within bounds.
PQTDAARevoke.LEA4 Secret k(LEA,CA)	Ok	No attacks within bounds.
PQTDAARevoke.LEA5 Secret h(vdh)	Ok	No attacks within bounds.
CA PQTDAARevoke.CA1 Secret k(LEA,CA)	Ok	No attacks within bounds.
PQTDAARevoke.CA2 Secret h(vdh)	Ok	No attacks within bounds.

Figure 6. Verification results for the PQ-TDAA-Pseudonym Changing, Resolution, and Revocation protocol using the Scyther tool, showing all security claims marked as Ok, with no attacks found within bounds.

The Scyther verification of the PQ-TDAA Network Registration (V2I) phase confirms that all modeled security goals are satisfied and that no attacks are detected. The origin (OBU) role achieves Alive, Nisynch, and Secret for both the pseudonym credential digest and the signed pseudonym, showing that an honest OBU completing the protocol interacts with a consistent, unique RSU run while keeping its credential data confidential from the Dolev–Yao adversary. On the responder (RSU) side, all Secret claims for auxiliary

data, the pseudonym digest, the signed pseudonym, and the underlying pseudonym are also verified, along with Nisynch and Niagree, which guarantee injective agreement and synchronization with the OBU for the registered credential tuple. These results indicate that the proposed Network Registration phase provides both strong mutual authentication and confidentiality of pseudonym-related information at the symbolic level. We also modeled the beacon broadcast, where a vehicle sends pse_{dig} , the RSU signature $sign_{pseu}$, and a TPM-signed beacon to nearby vehicles. Scyther reports failed authentication claims only in traces where the intruder played a vehicle role (e.g., runs the sender with a valid pseudonym) and thus corresponds to compromised OBU scenarios rather than a pure network attacker. These cases lie outside our threat model, which assumes honest vehicles and RSUs with uncompromised keys; under this assumption, only legitimate vehicles can produce beacons with valid RSU and TPM signatures.

The Scyther model for the Pseudonym Changing phase focuses on secrecy and consistency of the new pseudonym. The analysis confirms that $newpse_{digest}$ and $newpse_{cre}$ remain confidential against a Dolev–Yao network attacker and that the RSU always completes the protocol with parameters consistent with its own computations. Strong agreement claims on the vehicle side are not enforced since, by design, the RSU is the authoritative issuer that binds pseudonyms. In contrast, the vehicle only needs to verify the RSU’s signature and the integrity of the hybrid-encrypted response before adopting the new pseudonym.

The formal analysis of the Pseudonym Resolution and Revocation phases was performed using an automated protocol verification tool with the protocol specified in security protocol description language (SPDL), modeling three entities (RSU, LEA, and CA) and two symmetric session keys $k(RSU, LEA)$ and $k(LEA, CA)$. The verification results show that all relevant security claims are satisfied: (i) The LEA role fulfills the Alive claim, ensuring that every completed LEA run corresponds to honest executions of both RSU and CA; (ii) the Niagree and Nisynch claims between LEA and CA hold for the tuple $(psecre, h(VID_{in}), reason, T_q)$, which guarantees injective agreement and synchronization on the revocation record; and (iii) Secret claims for $k(RSU, LEA)$, $k(LEA, CA)$, and $h(VID_{in})$ at LEA and CA are also satisfied, indicating that both the session keys and the hashed vehicle identity remain confidential against a Dolev–Yao adversary.

4.2. Informal Analysis

We provide an informal justification of how PQ-TDAA satisfies eight security and privacy requirements defined based on our previous work in [3,42].

1. **Mutual and Anonymous Authentication:** The PQ-TDAA framework guarantees a high level of mutual authentication in the Join phase, where the OBU and CA should authenticate themselves with credentials by using encrypted messages and ZKP authentication. More importantly, the real vehicle identity (VID) remains hidden; later authentication in the Network Registration phase is performed via pseudonyms, with support from a simplified Schnorr with Fiat–Shamir transform ZKP (π_2). The process enables the vehicle to demonstrate legitimacy without disclosing its secrets, thus ensuring authenticated communication and maintaining complete sender anonymity and unlinkability.
2. **Data Integrity:** Data integrity is maintained based on a dual-tier PQC approach to unauthorized modification. The first layer provides non-repudiation and integrity for public messages using CRYSTALS-Dilithium2 signatures, ensuring that V2V beacons and pseudonym certificates can be verified and guaranteed. The second layer provides integrity for confidential communications (e.g., join, registration, and revocation) using the AES-256-GCM AEAD primitive. The primitive produces a cryptographically

secure authentication tag that enables the receiver to detect any modification and to quickly fail-stop reject an attacked message.

3. Confidentiality: The PQ-TDAA protocol achieves a high level of confidentiality at all sensitive stages, which is a quantum-resistant hybrid encryption architecture (KEM-DEM). The first step in this system uses the NIST standard ML-KEM-512 to securely encapsulate a symmetric secret key (K) with the recipient's public key, providing strong protection against quantum attacks. The high-speed AES-256-GCM AEAD then encrypts the actual data payload using the key K. Confidentiality is crucial in ensuring that initial registration secrets are kept, along with secrecy of the trace token (only known to LEA) and pseudonym credentials throughout the network stages, which eventually ensures that only the party with the appropriate private key can decrypt and verify the integrity of the data plus the secrets of initial registration.
4. Non-repudiation: Non-repudiation is satisfied using a systematic use of post-quantum secure digital signatures, namely Dilithium2 and Falcon-512. Because each critical transmission (e.g., pseudonym certificates and vital protocol messages) is digitally signed, the message's origin is uniquely bound to the sender's private key. This mechanism provides unquestionable cryptographic evidence of participation; that is, the signing party cannot rightfully deny having sent the message after the recipient has authenticated it.
5. Resistance Against Attacks: PQ-TDAA is resistant to attack: (i) Replay attack is addressed by adding timestamps and nonces to each critical message exchange, the Join and Network Registration, Broadcast Beacon, and Pseudonym Changing phases; (ii) combining mutual authentication (both OBU and RSU are verified) prevents man-in-the-middle attack; (iii) modification attacks are decisively avoided, as every pseudonym certificate and broadcast message is protected by a post-quantum secure digital signature (Dilithium2 and Falcon-512), allowing any recipient to verify the integrity and reject unauthorized alterations upon receipt; and (iv) quantum attacks are inherently resisted because all underlying cryptographic primitives, including commitments, ZKPs, and signatures, are lattice-based and align with the NIST PQC standards.
6. Complete Pseudonym Lifetime: This is a critical structural component of the PQ-TDAA protocol, which guarantees the privacy of users and conditional accountability over time. It is a lifecycle with five phases. Pseudonyms are created (issued) and then used for anonymous V2V and V2I communication. As a precautionary measure against user unlinkability, credentials are periodically changed (updated). The issue of accountability is addressed by enabling the LEA (trusted authorities) to resolve the pseudonym to the hash of the vehicle's internal identity (VIDin) when required. Finally, the lifecycle ends with the revocation phase, typically when the CA blacklists the vehicles after detecting malicious behavior.
7. Unlinkability: Unlinkability is one of the privacy objectives, ensuring that no third party can connect several pseudonyms or messages to the same vehicle across time. The PQ-TDAA protocol can secure this using a mixture of operational guidelines and cryptographic robustness. At the operational level, the system will occasionally replace old pseudonyms with new ones, and tracking old identities over time is not possible. PQ-TDAA uses the hiding property of lattice-based ZKP. During authentication, the OBU demonstrates the correctness of a pseudonym without necessarily revealing the secret value that identifies the relationships among identities.
8. Conditional Traceability: PQ-TDAA provides conditional traceability to strike a balance between the privacy of the users and the need to have accountability. Such responsibility is attained through the establishment of the LEA. This is determined

cryptographically in the first stage of the Join phase. Each pseudonym certificate is assigned a trace token that is securely encrypted under LEA's public key. As a result, upon identifying malicious behavior and obtaining legal permission, the LEA can use its private key to decrypt the token and retrieve the hashed vehicle identifier $H(\text{VID}_{\text{in}})$. The scenario also ensures that, although general privacy is maintained, accountability is maintained by performing a resolution that is controlled and secure.

4.3. Comparative Security and Privacy Requirements

Table 6 summarizes the comparison of the security and privacy requirements of V-LDAA [8] and PQ-TDAA. The existing scheme variant provides anonymous authentication, unlinkability, data integrity, and post-quantum security, but it does not support conditional traceability or a complete pseudonym lifecycle. In contrast, PQ-TDAA satisfies all the security and privacy requirements listed in Table 9.

Table 9. Comparison of Security and Privacy Requirements.

Security and Privacy Requirements	V-LDAA [8]	PQ-TDAA
Mutual authentication	No	Yes
Anonymous authentication	Yes	Yes
Data integrity	Yes	Yes
Confidentiality	No	Yes
Resistance against attack:		
a. Replay attack	No	Yes
b. Man-in-the-middle attack	No	Yes
c. Modification attack	No	Yes
d. Quantum attack	Yes	Yes
Complete pseudonym lifecycle	No	Yes
Unlinkability	Yes	Yes
Conditional traceability	No	Yes

4.4. Lattice-Based Security Analysis

This section provides a comprehensive security justification for PQ-TDAA's lattice-based blind signature, which adapts the simplified Schnorr ZKP with the Fiat–Shamir transform from Beullens et al. [13] for VANET constraints.

4.4.1. Soundness via Module-SIS Reduction

This subsection shows that forging a valid proof without knowing the secret is as hard as solving the M-SIS problem. This mathematically hard problem underlies NIST post-quantum standards like Dilithium and Falcon. We use the forking lemma to establish this connection. The verification equation in Algorithm A4 and Algorithm A5 provides the foundation for our security proof:

$$A \cdot z_s - B \cdot z_r = t - ch \cdot (C - \rho) \quad (2)$$

where A and B are public matrices, z_s and z_r are the prover's responses, t is the commitment value, ch is the challenge, C is the public commitment, and ρ represents the blinding factor. This equation must hold for any valid proof.

Theorem 1. (Soundness). *Let A be an adversary that produces a valid proof $\pi_2 = (t, z_s, z_r, ch)$ for the Create phase with probability ϵ in time T , making at most Q_H random oracle queries. Then there exists an algorithm B that solves $M\text{-SIS}_{\{n, m, q, \beta\}}$ with parameters $n = 1024$, $m = 1024$, $q = 7933$, and $\beta = \sqrt{\beta_s^2 + \beta_r^2} \approx 17,818$ in expected time $T' \approx 2T$ with probability $\epsilon' \geq \epsilon^2/Q_H - \text{negl}(\lambda)$. This means forging proofs is computationally equivalent to solving M-SIS, which is believed to be hard even for quantum computers.*

Proof of Theorem 1: Algorithm B simulates the protocol for A, programming the random oracle H. When A outputs a valid proof (t, z_s, z_r, ch) , B rewinds A to obtain a second valid proof (t, z'_s, z'_r, ch') with the same commitment t but different challenge $ch' \neq ch$ (probability ε/Q_H by the forking lemma). Subtracting the two verification equations:

$$A \cdot (z_s - z'_s) - B \cdot (z_r - z'_r) = (ch - ch') \cdot (C - \rho) \quad (3)$$

□

Since $ch, ch' \in \{-1, +1\}$ and $ch \neq ch'$, we have $ch - ch' \in \{-2, +2\}$, allowing B to extract the M-SIS solution:

$$v = [(z_s - z'_s) / (ch - ch'); -(z_r - z'_r) / (ch - ch')] \quad (4)$$

The extracted vector satisfies $[A \mid -B] \cdot v = (C - \rho) / (ch - ch')$ with norm $\|v\| \leq \sqrt{\beta_s^2 + \beta_r^2}$. In the worst case, where both $\|z_s - z'_s\| \leq 2\beta_s$ and $\|z_r - z'_r\| \leq r$, the M-SIS solution bound becomes $\beta = \sqrt{\beta_s^2 + \beta_r^2} \approx 17,818$. This reduction establishes that forging proofs without witness knowledge is computationally equivalent to solving M-SIS, which shares the same hardness assumption as NIST-standardized schemes such as Dilithium and Falcon.

4.4.2. Hiding via Module-LWE

This subsection explains how our commitment scheme hides secret information. Security relies on the M-LWE problem, which ensures that commitments appear completely random to an adversary. This prevents attackers from learning secrets or linking different protocol executions together.

Commitment binding (Algorithm A2): The commitment $c = B \cdot r + h$ binds to the secret r , where $B \in \mathbb{Z}_q^{1024 \times 1024}$ is sampled uniformly at random in Algorithm A1. Under the $M\text{-LWE}_{\{1024, 1024, 7933, \chi\}}$ assumption (where χ is the distribution induced by small-norm r), the value $B \cdot r$ is computationally indistinguishable from uniform random in $\mathbb{Z}_q^{1024}$. This means that even if an adversary sees many commitments, they cannot learn anything about the secret r or link commitments from the same vehicle across different protocol executions. This provides unlinkability, a critical privacy property for VANETs.

Credential commitment (Algorithm A4): The commitment $C = A \cdot s$, where s is the credential issued by the CA, provides computational hiding of s under the M-LWE assumption over the random matrix A . Our implementation uses direct Gaussian sampling to generate s rather than GPV trapdoor-based sampling. Security is maintained because the M-SIS-based ZKP, in subsequent phases, cryptographically enforces credential validity during verification. This approach offers simpler implementation while preserving the hiding property needed for anonymous authentication.

4.4.3. Parameter Selection and Security Level

We validate the security of our parameters through BKZ 2.0 lattice reduction analysis [44] and comparison with the original Beullens construction, which uses identical base parameters. Our scheme uses three fundamental parameters, each chosen to balance security with practical performance. First, lattice dimension $n = 1024$ provides strong security while keeping computation feasible for automotive hardware. Larger dimensions would be more secure but too computationally expensive for real-time VANET requirements; smaller dimensions would be faster but vulnerable to attacks. Second, modulus $q = 7933$ is a prime number that enables efficient 16-bit arithmetic operations (important for embedded processors) while being large enough to resist known lattice attacks. Third,

Gaussian width $\sigma = 232$ controls the randomness in our ZKP. This value ensures that the samples are statistically close to ideal distributions, which is necessary for the proofs to hide secret information properly.

The norm bounds $\beta_s = 8909$ and $\beta_r = 64$ represents a $1.2\times$ relaxation from Beullens' stricter bound of $\beta_s = 7424$ (calculated as $8909 \div 7424 = 1.2$). This adjustment trades a modest security margin for reduced rejection sampling overhead, which is critical for meeting VANET timing constraints.

The security of our protocol reduces to the M-SIS problem. An attacker attempting to forge proofs must find a short vector in a lattice of dimension $d = 2048$ with norm $\beta = \sqrt{\beta_s^2 + \beta_r^2} \approx 17,818$. We evaluate this hardness using the root Hermite factor $\delta \approx 1.00259$, which quantifies how well lattice reduction algorithms approximate the shortest vector in the lattice. Values closer to 1.0 indicate stronger attacks. Our $\delta \approx 1.00259$ means that practical attacks can only find vectors significantly longer than the actual shortest vector. Using standard cryptanalysis conversions [44], this root Hermite factor corresponds to a BKZ block size of approximately 728, meaning an attacker would need to run the BKZ lattice reduction algorithm with block size of 728 to break the scheme.

The computational cost of running BKZ with a block size of 728, accounting for quantum speedups via Grover's algorithm and conservatively adjusting for our 20% larger norm bound, yields an estimated quantum security level of approximately 125 bits. This estimate substantially exceeds NIST Level-1 requirements for post-quantum cryptography (≥ 64 -bit quantum security [45]), indicating that our parameter choices provide an adequate security margin despite the efficiency-oriented relaxation.

4.4.4. Rejection Sampling and Zero-Knowledge

Our scheme uses rejection sampling to ensure that the responses in proofs do not leak information about secret keys. This technique, introduced by Lyubashevsky [12], works by repeatedly sampling until the response follows the correct statistical distribution, independent of the secret.

The basic process works as follows. When generating a proof, the vehicle computes a response $z = y + ch \cdot \text{secret}$, where y is a random Gaussian sample and $ch \in \{-1, +1\}$ is the challenge from the verifier. The response is accepted only if its norm $\|z\|$ is below a threshold β . If the norm is too large, the response is rejected, and the process repeats with a new random sample y . This repetition continues until an acceptable response is found.

The efficiency of this process depends critically on the size of the secret. In the Join phase, the secret r is small ($\|r\| \approx 64$ from bound β_r in Table 7), requiring minimal repetition. With approximately 96% of samples accepted on the first attempt (since the rejection bound matches the expected norm), the repetition rate $M \approx 1.04$ means only 1.04 attempts are needed on average per successful proof, with negligible information leakage (statistical distance $\varepsilon \approx 0.04$).

In contrast, the credential s is much larger ($\|s\| \approx 2970$, roughly $\beta_s/3$ from Table 7). Direct rejection sampling would require too many attempts. Instead, we use computational zero-knowledge: The Fiat-Shamir transform sets $ch = H(\text{commitment} \parallel \text{message})$, making the challenge unpredictable until the commitment is published. Since attackers cannot predict the challenge in advance, they cannot exploit statistical patterns in the responses, even without perfect rejection sampling.

4.4.5. Trapdoor Construction and Credential Issuance

Unlike the original Beullens construction, which employs GPV trapdoor-based preimage sampling, PQ-TDAA deliberately adopts a Lyubashevsky-style framework [12], where credential issuance uses direct Gaussian sampling $s \leftarrow D_\sigma$ without a trapdoor structure.

This represents an alternative cryptographic paradigm validated by NIST's standardization of CRYSTALS-Dilithium.

In GPV-based schemes, the CA uses trapdoor T to compute credentials via NTRU inversion and fast Fourier transform (FFT)-based sampling. Lyubashevsky-style schemes sample credentials directly from discrete Gaussian distributions using random matrices without trapdoor knowledge, analogous to the distinction between NIST-standardized Falcon and Dilithium.

Security is preserved through three independent mechanisms, regardless of sampling method. First, the verification equation establishes M-SIS hardness via the forking lemma (Theorem 1). Second, commitment hiding derives from the M-LWE assumption over the random matrix A . Third, rejection sampling ensures zero-knowledge independent of the credential generation process. These mechanisms operate correctly regardless of whether credentials are generated via trapdoor sampling or direct Gaussian sampling.

This design eliminates NTRU trapdoor complexity, reduces memory requirements for OBU hardware, and aligns with NIST's primary signature standard. With identical base parameters to Beullens ($q = 7933$, $\sigma = 232$, $n = 1024$), PQ-TDAA achieves approximately 125 bits of quantum security, substantially exceeding NIST Level-1 requirements (≥ 64 bits). The M-SIS norm $\beta \approx 17,818$ yields root Hermite factor $\delta \approx 1.00259$, well below the threshold of 1.0045 [44]. The 20% relaxation in norm bounds provides substantial implementation advantages in automotive environments while maintaining strong security.

4.4.6. Side-Channel Resistance

Our implementation incorporates specific countermeasures to mitigate timing-based side-channel leakage while maintaining computational efficiency. Three primary countermeasures are applied. First, discrete Gaussian sampling utilizes a precomputed cumulative distribution table (CDT) with binary search, replacing variable-time rejection sampling with table-based lookup to reduce timing variations. Second, to prevent timing analysis of the rejection sampling loops, we enforce strict, fixed iteration limits (200 for Join, 500 for Create). This provides a deterministic upper bound on execution time, independent of secret nonce values. Finally, the verification logic in Algorithm A2, Algorithm A3, Algorithm A4 and Algorithm A5 employs Euclidean L2 norm checks ($\|z\|_2 \leq \beta$) using NumPy's BLAS-accelerated `numpy.linalg.norm` (OpenBLAS 3.9.0 with AVX2), guaranteeing consistent processing time regardless of vector magnitude.

5. Performance Evaluation

This section evaluates PQ-TDAA through comparison with V-LDAA [8] and experimental benchmarking of Dilithium2 and Falcon-512. The evaluation covers security requirements, computational and communication costs, and network performance using NS-3 simulations. Consistent with SAE J2945/1 [46], scalability considerations in VANET safety systems primarily arise from broadcast-based V2V communications, where vehicles periodically transmit signed messages to all neighboring nodes. In this setting, scalability is dominated by signature size and verification latency, since safety beacons carry publicly observable content and use signature-only authentication, whereas ML-KEM-512 is used only in infrequent phases and does not affect broadcast performance. Network performance is assessed using packet delivery ratio (PDR), end-to-end delay (< 100 ms), authentication delay, and goodput.

5.1. Experimental Setup

Computation time was measured using SageMath 10.6 for lattice primitives and Python 3.11 for protocol-level timing. All experiments were executed on the hardware and software platform summarized in Table 10.

Table 10. Experimental Hardware and Software Environments.

Components	Specification
Hardware Environment:	
Device	ASUS ROG Strix G513QM laptop
Processor	AMD Ryzen 9 5900HX with Radeon graphics (8 cores, 16 threads, 3.30 GHz)
Installed RAM	32 GB DDR4, 3200 MT/s
Graphics card	NVIDIA GeForce RTX 3060
Storage	954 GB SSD
Operating system	Windows 11 Pro (64-bit) via WSL2 (Ubuntu 24.04.3 LTS, Linux kernel 6.6.87.2)
Software Environment:	
Mathematical framework	SageMath 10.6 kernel (Python 3.11.13) for lattice-based operations
Execution platform	Jupyter notebook (single-threaded configuration)
PQC runtime environment	Python 3.11 with liboqs 0.14.1-dev and liboqspython 0.14.1 bindings
Supported PQC algorithms	ML-KEM (Kyber512), CRYSTALS-Dilithium2, Falcon-512

5.2. Theoretical Computation Cost

The computational complexity of the proposed PQ-TDAA framework is determined by two distinct cryptographic building blocks: lattice-based blind signatures and NIST PQC standard algorithms. To assess the theoretical computational cost, we identified the cryptographic building-block cost of PQ-TDAA and measured the time required for each building block. Tables 11 and 12 present the building blocks of cryptography in SageMath and Python implementation, respectively, along with their computational time analysis.

Table 11. Computation Time of Building Block Cryptography in SageMath Implementation.

Building Block Cryptography	Description	Computation Time (μ s)	Notation
Matrix-vector multiplication	Performs matrix-vector multiplication in $\mathbb{Z}_q^{m \times n}$ (e.g., computing $Az_s - Bz_r$).	929.66	C_{matvec}
Gaussian sampling	Samples a length- n discrete Gaussian vector (σ) to generate secrets/perturbations (f, g, y, s) for trapdoor/ZKP algorithms.	7360.31	C_{gauss}
Uniform sampling	Uniformly samples small-element vectors (e.g., $\{-2, \dots, 2\}^n$), used to generate secret r in the Join phase.	607.85	C_{uniform}
L2 norm	Computes Euclidean norm $\ v\ _2$ for lattice vectors; used in rejection sampling and validating r, z, s bounds.	82.21	C_{L2norm}
Fast SHA-256	Optimized SHA3-256 execution for short, fixed inputs (ID, commitment, nonce ciphertext) without string processing overhead.	0.66	C_{fasthash}
AEAD-like (XOR + hash)	Lightweight primitive: Generates SHAKE keystream for XOR encryption and computes SHA3 tag over nonce/ciphertext for integrity.	99.26	$C_{\text{AEAD-like}}$

Table 12. Computation Time of Building Block Cryptography in Python Implementation.

Building Block Cryptography	Description	Computation Time (μs)	Notation
ML-KEM key generation	Generates an ML-KEM-512 key pair (public and secret keys).	18.1	$C_{\text{ML-KEM keygen}}$
ML-KEM encapsulation	Uses ML-KEM public key to generate KEM ciphertext and the shared secret (AES-GCM session key).	13.9	$C_{\text{ML-KEM encap}}$
ML-KEM decapsulation	Uses ML-KEM secret key to recover the shared secret from the KEM ciphertext on the receiver's side.	14.6	$C_{\text{ML-KEM decap}}$
Dilithium2 key generation	Generates a public and secret key pair for digital signatures using Dilithium2.	30.3	$C_{\text{Dilithium2 keygen}}$
Dilithium2 signing	Signs a message using the Dilithium2 secret key.	50.8	$C_{\text{Dilithium2 sign}}$
Falcon-512 key generation	Generates a public and secret key pair for digital signatures using Falcon-512.	4703.2	$C_{\text{Falcon-512 keygen}}$
Falcon-512 signing	Signs a message using the Falcon-512 secret key.	182.9	$C_{\text{Falcon-512 sign}}$
Falcon-512 verify	Verifies a Falcon-512 signature using the associated public key.	47.1	$C_{\text{Falcon-512 verify}}$
AEAD encrypt	AES-256-GCM encryption using the ML-KEM symmetric key; generates nonce, ciphertext, and authentication tag.	2.3	$C_{\text{AEAD encrypt}}$
AEAD decrypt	AES-256-GCM decryption; verifies the auth tag and returns plaintext upon successful authentication.	1.6	$C_{\text{AEAD decrypt}}$
SHA3-256	Generates protocol digests (psdigest, omega/theta, h(VID)) using SHA3-256 hashing.	2.1	$C_{\text{SHA3-256}}$

Table 13 details the theoretical performance of the PQ-TDAA framework, calculated by summing the weighted costs of constituent cryptographic primitives for Dilithium2 and Falcon-512. The analysis reveals that the Join and Network Registration phases account for most of the computational overhead due to their heavy reliance on lattice ZKPs, KEMs, and signature operations, whereas Resolution and Revocation remain lightweight.

Table 13. Theoretical Computational Time of the PQ-TDAA Framework with Dilithium2 and Falcon-512 Implementation (in seconds).

Phase	Theoretical Computation Expression	PQ-TDAA-Dilithium2	PQ-TDAA-Falcon-512
Setup	$2C_{\text{gauss}} + 4C_{\text{ML-KEM keygen}} + (4C_{\text{Dilithium2 keygen}} \text{ or } 4C_{\text{Falcon-512 keygen}})$	0.0149	0.0336
Join	$4C_{\text{matvec}} + 3C_{\text{gauss}} + C_{\text{uniform}} + 5C_{\text{L2norm}} + 8C_{\text{fasthash}} + 2C_{\text{AEAD-like}} + 5C_{\text{ML-KEM encap}} + 4C_{\text{ML-KEM decap}} + (2C_{\text{Dilithium2 sign}} \text{ or } 2C_{\text{Falcon-512 sign}}) + (2C_{\text{Dilithium2 verify}} \text{ or } 2C_{\text{Falcon-512 verify}}) + 5C_{\text{AEAD encrypt}} + 4C_{\text{AEAD decrypt}} + 3C_{\text{SHA3-256}}$	0.0273	0.0276
Create	$3C_{\text{matvec}} + 2C_{\text{gauss}} + 2C_{\text{L2norm}} + 2C_{\text{fasthash}}$	0.0178	0.0180

Table 13. Cont.

Phase	Theoretical Computation Expression	PQ-TDAA-Dilithium2	PQ-TDAA-Falcon-512
Network Registration (V2I)	$2C_{matvec} + 2C_{L2norm} + C_{fasthash} + 2C_{encap}^{ML-KEM} +$ $2C_{decap}^{ML-KEM} + (C_{sign}^{Dilithium2} \text{ or } C_{sign}^{Falcon-512}) + (C_{verify}^{Dilithium2}$ atau $C_{verify}^{Falcon-512}) + 2C_{encrypt}^{AEAD} + 2C_{decrypt}^{AEAD} + 2C_{SHA3-256}$	0.0020	0.0020
Broadcast Beacon (V2V)	$(C_{sign}^{Dilithium2} \text{ or } C_{sign}^{Falcon-512}) + (2C_{verify}^{Dilithium2} \text{ or } 2C_{verify}^{Falcon-512})$	0.0001	0.0003
Pseudonym Changing	$2C_{encap}^{ML-KEM} + 2C_{decap}^{ML-KEM} + (C_{sign}^{Dilithium2} \text{ or } C_{sign}^{Falcon-512}) + (C_{verify}^{Dilithium2} \text{ or } C_{verify}^{Falcon-512}) + 2C_{encrypt}^{AEAD} + 2C_{decrypt}^{AEAD} + C_{SHA3-256}$	0.0001	0.0003
Pseudonym Resolution	$C_{encap}^{ML-KEM} + 2C_{decap}^{ML-KEM} + (C_{verify}^{Dilithium2} \text{ or } C_{verify}^{Falcon-512}) + C_{encrypt}^{AEAD} + 2C_{decrypt}^{AEAD}$	0.0001	0.0001
Pseudonym Revocation	$C_{encap}^{ML-KEM} + C_{decap}^{ML-KEM} + (C_{verify}^{Dilithium2} \text{ or } C_{verify}^{Falcon-512}) + C_{encrypt}^{AEAD} + C_{decrypt}^{AEAD}$	0.0001	0.0003
TOTAL		0.0625	0.0822

Overall, PQ-TDAA-Dilithium2 outperforms the Falcon variant, with a total cycle time of 0.0625 s versus 0.0822 s. The increased latency in PQ-TDAA-Falcon-512 is due to the higher computational burden of Falcon-512's key generation and signing. These estimates provide a critical analytical baseline for comparison against the subsequent experimental benchmarks in SageMath and Python.

5.3. Asymptotic Complexity Analysis and Feasibility Analysis

To validate the PQ-TDAA framework's scalability, we correlated the precise arithmetic operation counts derived in Section 5.3 with a generalized asymptotic complexity analysis. The results, summarized in Table 14, demonstrate a strategic architectural trade-off between privacy assurance and real-time performance.

Table 14. Asymptotic Complexity of the PQ-TDAA Framework.

Phase	Dominant Operation	Dominant Complexity
Setup	Matrix generation $A, B \in \mathbb{Z}_q^{m \times n}$	$O(mn)$
Join	Blind signature issuance (matrix-vector multiplication)	$O(mn)$
Create	Commitment computation and ZKP generation	$O(mn)$
Network Registration (V2I)	ZKP verification $Az_s - Bz_r$	$O(mn)$
Broadcast Beacon (V2V)	Sign and verify using Dilithium2 or Falcon-512	$O(k^2 n \log n)$
Pseudonym Changing	Sign and verify using Dilithium2 or Falcon-512	$O(k^2 n \log n)$
Pseudonym Resolution	Verify using Dilithium2 or Falcon-512 and KEM with AEAD	$O(k^2 n \log n)$
Pseudonym Revocation	Sign and verify using Dilithium2 or Falcon-512 and KEM with AEAD	$O(k^2 n \log n)$

1. Privacy-centric initialization (Setup to V2I): To guarantee robust anonymity, these phases employ lattice-based blind signatures and ZKPs over unstructured lattices. The dominant matrix operations ($A \in \mathbb{Z}_q^{m \times n}$) result in $O(mn)$ complexity, justifying the higher computational cost observed during the registration phase.

2. Latency-critical operation (V2V to Revocation): Once registered, the framework transitions to optimized module lattice primitives (e.g., Dilithium/Falcon). By leveraging the algebraic structure of polynomial rings and the number theoretic transform, the complexity is reduced to $O(k^2 n \log n)$, ensuring the low latency required for high-frequency beaconing.

Despite the higher asymptotic complexity during initialization, the framework remains strictly feasible for VANET environments due to amortized efficiency. The computationally intensive registration cost ($O(mn)$) is incurred only infrequently as a transient load, which modern OBUs can handle without compromising safety. Conversely, the high-frequency safety beacons leverage lightweight ($O(k^2 n \log n)$) operations that execute in the sub-millisecond range, falling well within the 100 ms latency budget mandated by safety standards such as IEEE 1609.x, thus ensuring strictly real-time performance.

5.4. Experimental Computation Cost

To validate the theoretical efficiency and feasibility discussed in Section 5.3, we implemented the PQ-TDAA framework and measured the cryptographic execution time for each stage. Table 15 shows the experimental computation-time estimates for PQ-TDAA with Dilithium2 and Falcon-512 implementations.

Table 15. Experimental Computation Time Estimation of PQ-TDAA with Dilithium2 and Falcon-512 Implementation (in seconds (s)).

Phase	PQ-TDAA	
	Dilithium2	Falcon-512
Setup	0.0082	0.0304
Join	0.0427	0.0430
Create	0.0269	0.0269
Network Registration (V2I)	0.0117	0.0120
Broadcast Beacon (V2V)	0.0012	0.0014
Pseudonym Changing	0.0012	0.0009
Pseudonym Resolution	0.0003	0.0006
Pseudonym Revocation	0.0001	0.0003
TOTAL	0.0926	0.1156

The experimental results comparing PQ-TDAA-Falcon-512 and PQ-TDAA-Dilithium2 reveal a simple trade-off that often defines real-world network security: speed versus size. Our low-level tests confirm that Dilithium2 is generally faster at individual cryptographic steps, such as key generation, message signing, and signature verification. However, because both schemes operate at extremely high speeds—with time differences measured in microseconds that barely register when converted to seconds—this speed advantage becomes marginal in the protocol's larger context. In fact, during the critical Broadcast Beacon (V2V) phase, where vehicles communicate continuously, both schemes are practically equivalent, demonstrating they are fast enough for real-time vehicular traffic.

5.5. Blind Signature Comparison with V-LDAA

In this subsection, we provide an architectural and implementation-level comparison between the blind signature mechanisms of V-LDAA [8] and PQ-TDAA in a unified experimental environment. To gain practical insights into V-LDAA's design, we developed a simplified research prototype following Chen et al.'s [8] parameters ($d = 128$, $q = 114,356,107$, and $\sigma = 232$) in SageMath with number theory library (NTL) acceleration. Our prototype employs real discrete Gaussian sampling via a CDT-based sampler and rejection sampling with norm bounds, but uses simplified ZKP instantiations (basic Fiat–Shamir transformation) rather than the complete Stern-type protocols required for full V-LDAA

security. This prototype illustrates computational patterns and is not a faithful V-LDAA implementation. For authoritative comparison, we rely on measurements reported in the original V-LDAA paper [8].

V-LDAA [8] employs an optimized signature scheme based on the automorphism stability of power-of-two cyclotomic fields, reducing the number of polynomials from 40 to 36 by proving only σ_5 automorphism stability (removing the σ_{-1} proof). The authentication credential PSCert comprises three components: the pseudonym public key (nym), the revocation certificate ($\text{sig}_1 = H_{\text{Rq}}(\text{nym}, e)$), and the blind signature sig_2 , which contains the 36-polynomial ZKP structure. These polynomials are distributed across multiple parallel proofs: π_1' for commitment opening, π_2' for automorphism stability verification, π_3' for credential validity, and TPM binding components. Using V-LDAA's parameters ($d = 128$, $q \approx 114,356,107$, requiring approximately 4 bytes per coefficient), the 36-polynomial structure yields approximately 18 KB for the core blind signature component (36×512 bytes per polynomial). The complete authentication credential reported in [8] is approximately 26 KB. This suggests an additional overhead of approximately 8 KB attributed to pseudonym, revocation certificate, protocol metadata, and encoding, although a detailed component-level breakdown is not provided in the original paper. For our comparison, we used the authoritative 26 KB measurement from [8].

For PQ-TDAA, we instantiated the Beullens' lattice blind signature construction [13] with simplified Schnorr and Fiat–Shamir transform, using parameters $n = m = 1024$, $q = 7933$, and $\sigma = 232$, and measured the cost of the blind signing and blind verification algorithms using the same benchmarking harness. The blind signature comprises a pseudonym ρ (2048 bytes) and ZKP π_2 (6145 bytes containing commitment t , responses z_r and z_s at 2048 bytes each, plus binary challenge $ch \in \{-1, +1\}$), totaling 8193 bytes.

Table 16 reports the computation time and proof size for the blind signature operations of V-LDAA [8] and PQ-TDAA. For V-LDAA [8], our simplified but structurally faithful implementation yields 7.26 ms for BlindSign and 0.14 ms for BlindVerify, with a proof size of 25.50 KB. These timing results are significantly faster than the 5900 ms BlindSign and 47 ms BlindVerify reported in [8] due to our use of simplified ZKP instantiations (fundamental Fiat–Shamir transformation) rather than the complete Stern-type proofs employed in the original V-LDAA. Critically, the signature size of 26 KB accurately reflects V-LDAA's complete PSCert structure as reported in the original paper, ensuring a fair architectural comparison. In contrast, the PQ-TDAA records 22.45 ms for BlindSign and 8.21 ms for BlindVerify, with the proof size fixed at 8 KB as determined by the underlying Beullens-style Schnorr proof construction.

Table 16. Experimental Comparison of Blind Signature Operations between V-LDAA [8] and PQ-TDAA.

Scheme	BlindSign Time (ms)	BlindVerify Time (ms)	Blind Proof Size (KB)	Size Reduction
V-LDAA [8]	7.26	0.14	26	-
PQ-TDAA	22.45	8.21	8	69.2%

The results highlight a clear trade-off between local computation and communication overhead in post-quantum VANET authentication. Although V-LDAA's blind signing and verification are faster in this prototype implementation due to simplified ZKP, its 26 KB authentication credentials are more than three times larger than PQ-TDAA's 8.0 KB credentials. This size difference is architecturally fundamental: V-LDAA's automorphism-based construction requires 36 polynomials for ZKP over lattice elements, while PQ-TDAA's Schnorr-based approach requires only five compact components (ρ and π_2 (t , z_r and z_s),

ch)). These authentication credentials are transmitted during the Network Registration phase, when vehicles obtain or refresh their pseudonymous certificates, so their size directly influences bandwidth consumption when many OBUs register concurrently.

5.6. Real ARM Cortex-A76 V2V Measurements

Section 5.4 establishes the AMD Ryzen 9 5900HX workstation baseline for PQ-TDAA performance. Building on the $3.26\times$ theoretical scaling projected in Table 2 (Section 2.1.4), Section 5.6 presents direct measurements on ARM Cortex-A76 using Raspberry Pi 5 (BCM2712 quad-core @2.4 GHz), a realistic automotive OBU platform. These measurements capture the complete V2V cycle, representing actual cryptographic workload in VANET beaconing; that is, beacon generation on the sender side includes the time required to construct a safety message and generate a digital signature (sign), and beacon verification on the receiver side consists of the time needed to validate a received message from a neighboring vehicle (verify). There are two processes of verification: verification of the RSU's signature and the beacon's signature. Table 17 compares raw performance across both platforms.

Table 17. Measured V2V Performance: Ryzen vs. Cortex A-76.

V2V Sub-Process	Operation	Our Workstation (Ryzen 9)	R-Pi 5 Cortex A-76	Scaling
		((Measured (ms))	(Measured (ms))	
Dilithium2				
Beacon generation	Beacon pack	0.0078	0.0134	3.08 times (95% of 3.26 times)
	Beacon signing	0.0670	0.2085	
	Total	0.0748	0.2219	
Beacon verification	Verification	0.0524	0.1699	
Total V2V Cycle		0.1272	0.3918	
Falcon-512				
Beacon generation	Beacon pack	0.0062	0.0089	1.59 times (49% of 3.26 times)
	Beacon signing	0.1910	0.3080	
	Total	0.1972	0.3169	
Beacon verification	Verification	0.0787	0.1199	
Total V2V Cycle		0.2759	0.4368	

We observed an architecture-specific effect. We found that Dilithium performed faster on ARM processors because its integer-based mathematical operations naturally parallelize across NEON's single instruction multiple data processing, which handles four coefficients simultaneously [47]. This results in a $3.08\times$ speedup (95% of the theoretical maximum). Falcon achieved a lower $1.59\times$ speedup (49% of theoretical) because its complex floating-point FFT operations require specialized manual NEON optimization for full performance [48], as shown by just 16.5% additional signing speedup with such tuning. Both schemes deliver sub-0.5 ms V2V cycles, well within the 100 ms beacon budget, confirming real-time suitability for vehicle OBU.

5.7. Consistency Analysis: Theoretical vs. Experimental Computation

In this section, we validate the integrity of the proposed PQ-TDAA framework by comparing the theoretical computational costs reported in Table 13 with the experimental execution times reported in Table 15. This end-to-end comparison demonstrates that the

mathematical model accurately predicts the system's behavior across all lifecycle phases, from initialization to revocation.

Figure 7 shows a side-by-side comparison of execution times across lifecycle phases for the Dilithium2 and Falcon-512 implementations.

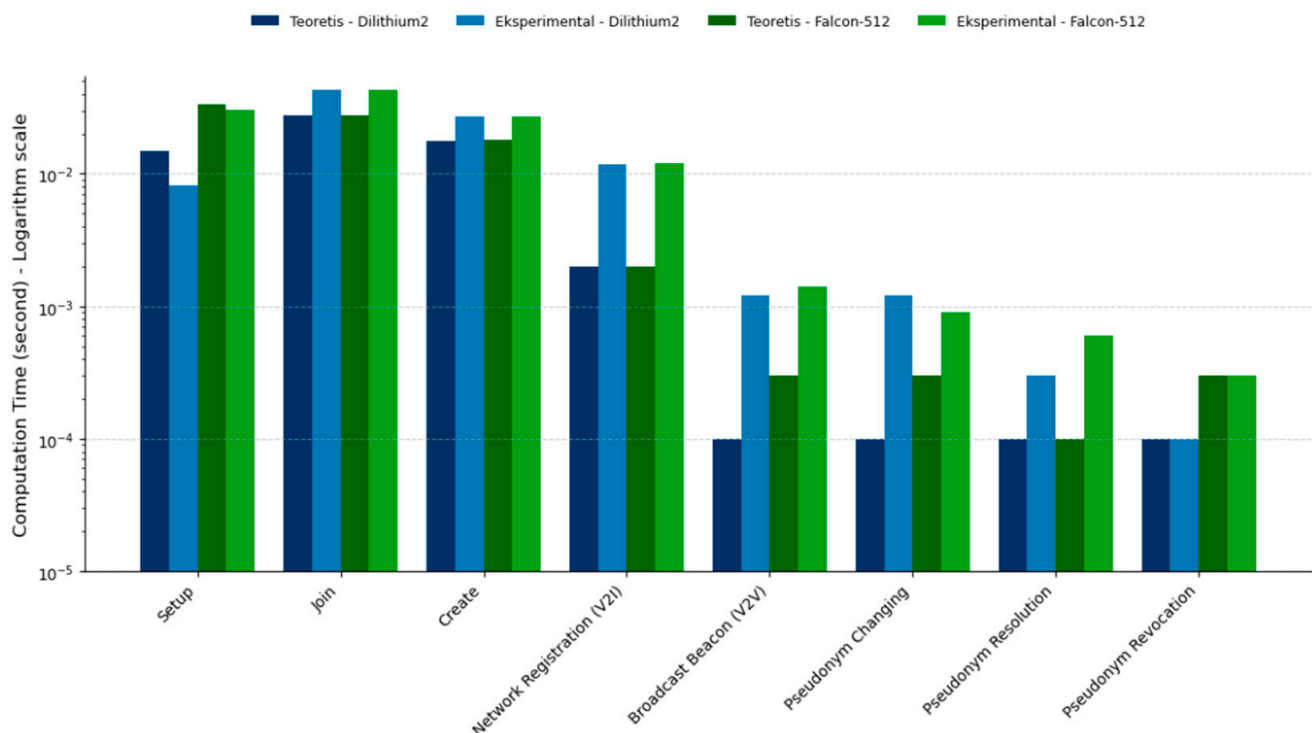


Figure 7. Comparison of Theoretical Predictions and Empirical Measurements for Dilithium2 and Falcon-512 Performance.

The consistency analysis between theoretical computational costs and experimental execution times confirms the high reliability of the PQ-TDAA framework. As illustrated in the logarithmic comparison in Figure 7, empirical results for both Dilithium2 and Falcon-512 closely follow the trends predicted by the mathematical model derived in Table 12. For Dilithium2, the measured total system time of 0.0868 s aligns with the theoretical estimate of 0.0625 s, while Falcon-512 shows a similar correlation with an experimental total of 0.1059 s against a predicted 0.0822 s.

Phase-specific analysis further validates the design's efficiency, particularly in the safety-critical Broadcast Beacon (V2V) phase, where the negligible theoretical costs (0.0001–0.0003 s) are mirrored by ultra-low experimental latencies (0.0013–0.0015 s). While complex operations such as Join and Create exhibit slight experimental overhead due to practical systemic factors such as memory access and operating system scheduling, the Setup phase for Dilithium2 ran notably faster in practice (0.0080 s) than the conservative theoretical bound (0.0149 s). Ultimately, the strong correlation across all lifecycle phases confirms that the proposed framework successfully translates algebraic efficiency into practical real-time performance for vehicular networks.

5.8. Communication Cost Analysis

The communication cost analysis focuses on the Broadcast Beacon phase, which accounts for more than 95% of the communication overhead. The message in the Broadcast Beacon phase of the PQ-TDAA framework consists of $pse_{digest} \parallel sign_{pseu}, sign_{msg}, beacon, pk_{TPM}^{DSA}$ and T_c . Table 17 shows the component sizes for the PQ-TDAA implementation using Dilithium2 and Falcon-512.

Table 18 shows that the communication cost during the Broadcast Beacon phase depends on the signature size. Both variants share small and fixed-size elements such as the pseudonym hash, beacon payload, and timestamp. However, the signature fields account for the majority of the total cost. Dilithium2 produces large signatures (approximately 2.4 KB each), resulting in an overall message size of 6254 bytes. Falcon-512, in contrast, generates much more compact signatures (approximately 666 bytes), reducing the total to 2331 bytes, approximately $2.7\times$ smaller.

Table 18. Communication Cost of PQ-TDAA with Dilithium2 and Falcon-512 Implementation (in bytes).

Component	Description	PQ-TDAA	
		Dilithium2	Falcon-512
pse_{digest}	Hash result of the pseudonym credential	32	32
$sign_{pseu}$	Signature of pseudonym pse_{digest}	2420	666
$sign_{msg}$	Signature of beacon	2420	666
beacon	(pos speed notif)	64	64
pk_{TPM}^{DSA}	Public key DSA of the vehicle	1312	897
T_c	Timestamp	6	6
TOTAL		6254	2331

5.9. Network Performance Evaluation

This network performance analysis examines the effects of these framework-level efficiencies on the overall network performance in real vehicular scenarios. A decentralized V2V broadcasting environment is simulated in NS-3, where each vehicle periodically transmits authenticated beacon messages to its neighbors. It measures four key performance metrics—PDR, average end-to-end (E2E) delay, average authentication delay, and goodput—to determine the reliability, latency, and bandwidth efficiency of every scheme under different traffic loads. Both the signing and verification delay models depend on the experimental data of the measured execution time of the Broadcast Beacon (V2V) phase. The experimentally derived delays are injected into NS-3 as deterministic scheduling parameters, enabling the simulator to model realistic per-packet cryptographic latency without executing the actual post-quantum primitives. Table 19 presents the sign-and-verify computation time for PQ-TDAA using the Dilithium2 and Falcon-512 implementations.

Table 19. Sign and Verify Computation Time Used in NS-3 Simulation.

Scheme	Sign (μ s)	Verify (μ s)
PQ-TDAA-Dilithium2	66.9184	52.4420
PQ-TDAA-Falcon-512	190.9926	78.7414

5.9.1. Network Simulation Parameter

The NS-3 simulation uses realistic IEEE 802.11p parameters: Nakagami-m fading ($m_0 = 4.0$, $m_1 = 4.0$, $m_2 = 3.0$), CCA threshold (-82 dBm), and RxSensitivity (-85 dBm at 6 Mbps orthogonal frequency division multiplexing (OFDM)). These parameters capture vehicular multipath effects while isolating PQ-TDAA's cryptographic overhead from higher-layer network dynamics. Two deliberate simplifications remain: no log-normal shadowing (per 3GPP TR 36.885) and no lane-changing mobility. These omissions create a conservative baseline. Shadowing would increase retransmissions due to signal obstructions (e.g., trucks or buildings), thereby elevating the cryptographic load. Lane changes would trigger more frequent neighbor discoveries and verification operations. Both effects would degrade

PDR and increase delay beyond current measurements, making the reported performance a realistic lower bound.

The main parameters used in the NS-3.38 simulation are summarized in Table 20. The simulations use a 300 m road segment with four vehicle density scenarios: 10 vehicles (33.3 veh/km) representing free-flow traffic (LOS B-C), 20 vehicles (66.7 veh/km) representing approaching unstable flow (LOS D), 50 vehicles (166.7 veh/km) representing unstable flow (LOS E), and 100 vehicles (333.3 veh/km) representing stress-test breakdown flow (LOS F) per Indonesian highway classification [49].

Table 20. NS-3 Simulation Configuration Parameters.

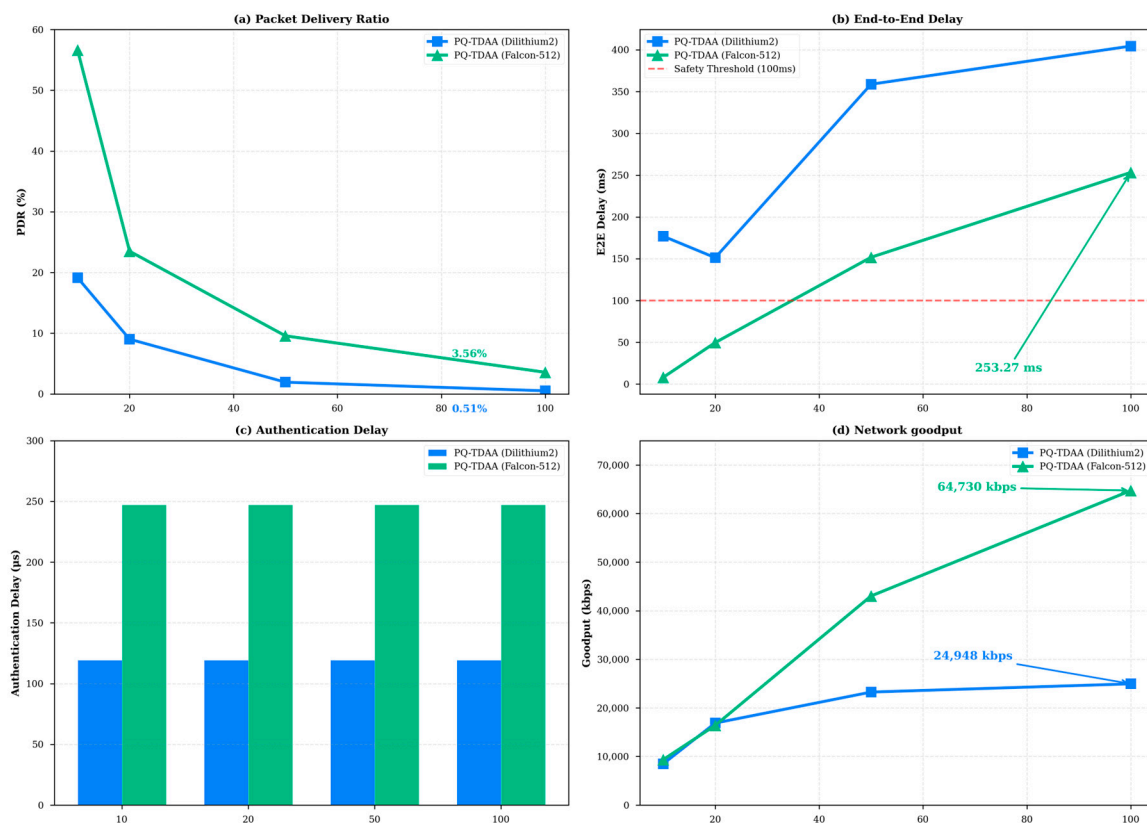
Parameter	Description/Value
Simulation time	200 s
Number of vehicles	10, 20, 50, and 100
Road length	300 m
Mobility model	Constant velocity mobility model
Beacon interval	100 ms
Transmission range	300 m
Vehicle speed range	15–25 m/s
MAC standard	IEEE 802.11 p
Transmit power	33 dBm
Propagation	Nakagami-m fading ($m_0 = 4.0$, $m_1 = 4.0$, $m_2 = 3.0$)
CCA threshold	−82 dBm (802.11 p standard)
Rx sensitivity	−85 dBm (6 Mbps OFDM)
Data rate	6 Mbps OFDM (10 MHz channel)
Shadowing	None (future: 3GPP TR 36.885)
Mobility	Constant velocity (no lane changes)

The $N = 100$ case specifically validates protocol behavior under MAC-layer collapse (severe contention and fragmentation dominance). This extreme scenario complements operational densities ($N = 10$ –50) representing LOS B-E conditions, providing a comprehensive evaluation across operational and stress regimes. Such density levels align with standard VANET evaluation practices that use tens to several hundred vehicles to assess scalability and broadcast reliability, as demonstrated by Karnadi et al. [50] and Sommer et al. [51]. This alignment ensures that PQ-TDAA is evaluated across realistic, literature-consistent operating ranges.

Table 21 and Figure 8 present the results of NS-3 simulations for PDR, average E2E delay, average authentication delay, and network goodput of PQ-TDAA with Dilithium2 and Falcon-512 implementations. In the NS-3 simulations, the proof size was integrated into the pseudonym certificate rather than treated as an isolated payload, since the V2V beacon message in VANET communication inherently includes multiple cryptographic components—specifically, 6254 bytes for PQ-TDAA with Dilithium2 and 2331 bytes for PQ-TDAA with Falcon-512. Therefore, comparisons of end-to-end and authentication delays reflect the overall impact of the beacon composition rather than the proof size alone. This approach ensures a realistic evaluation of communication latency, as each beacon encapsulates the RSU-validated pseudonym certificate, the vehicle’s public keys, and the RSU’s digital signatures.

Table 21. Comparison of Network Performance Across Different Vehicle Densities of PQ-TDAA (Dilithium2 and Falcon-512) Variants.

Scenario	Vehicle = 10	Vehicle = 20	Vehicle = 50	Vehicle = 100
PQ-TDAA (Dilithium2)				
Beacon sent	19,700	39,400	98,500	197,000
Total receptions	33,905	67,446	92,950	99,729
PDR (%)	19.12	9.01	1.93	0.51
Avg E2E delay (ms)	176.987	151.296	358.781	404.464
Avg auth delay (μ s)	119.360	119.360	119.360	119.360
Goodput (kbps)	8481.67	16,872.29	23,252.37	24,948.21
PQ-TDAA (Falcon-512)				
Beacon sent	19,700	39,400	98,500	197,000
Total receptions	100,342	175,818	461,472	694,235
PDR (%)	56.59	23.49	9.56	3.56
Avg E2E delay (ms)	8.127	49.699	151.782	253.273
Avg auth delay (μ s)	269.734	269.734	269.734	269.734
Goodput (kbps)	9355.89	16,393.27	43,027.65	64,730.47

**Figure 8.** Overall V2V network performance comparison PQ-TDAA variants under varying vehicle densities: (a) packet delivery ratio (PDR), (b) end-to-end (E2E) delay, (c) authentication (auth) delay, and (d) network goodput.

5.9.2. Packet Delivery Ratio (PDR) Analysis

Table 21 and Figure 8a show PDR results and performance across varying vehicle densities. Following the broadcast evaluation methodology of [52,53], PDR is calculated as:

$$PDR (\%) = \frac{\text{Total receptions}}{\text{Beacon sent} \times (N - 1)} \times 100 \quad (5)$$

where N denotes the number of vehicles and N – 1 represents the potential neighbors for each broadcast beacon.

At low density ($N = 10$), PQ-TDAA-Falcon-512 achieves 56.59% PDR compared to Dilithium2's 19.12%, representing a 2.96-fold improvement. This advantage widens as the network becomes more congested; at $N = 100$, Falcon-512 still preserves a PDR of 3.56%, whereas Dilithium2 drops to 0.51%, yielding roughly a seven-fold improvement despite the overall degradation in reliability under extreme load. Falcon-512 maintains a $3\text{--}7\times$ PDR advantage across all densities ($56.59\% \rightarrow 3.56\%$ vs. $19.12\% \rightarrow 0.51\%$), demonstrating that a $4.4\times$ reduction in beacon size (2331 B vs. 6254 B) translates directly into superior network scalability under IEEE 802.11p constraints. The performance gap is primarily driven by fragmentation: Constrained by the IEEE 802.11p maximum transmission unit (MTU) of 1500 bytes, Dilithium2 beacons of 6254 bytes are divided into five IP fragments, whereas Falcon-512 beacons of 2331 bytes require only two fragments, and successful delivery depends on all fragments arriving without collision.

In the stress-test regime ($N = 100$, ≈ 333 vehicles/km), the PDR of PQ-TDAA-Falcon-512 falls to 3.56% due to severe MAC-layer saturation, where the order of 3000 broadcast messages per second exceeds the adequate IEEE 802.11p channel capacity under breakdown-flow traffic. This behavior is consistent with expectations for heavily congested VANET links: The observed packet loss is a consequence of channel overload rather than a limitation of the authentication mechanism, since PQ-TDAA continues to provide complete cryptographic integrity, and every delivered beacon carries a valid, verifiable signature. From a safety perspective, the operating region up to $N = 50$ vehicles, where Falcon-512 maintains PDR values above 45%, corresponds to typical urban and peri-urban densities where beacon reception remains dependable. In contrast, more aggressive congestion control or beacon-rate adaptation is required only under rare breakdown-flow conditions.

5.9.3. End-to-End (E2E) Delay Analysis

Table 21 and Figure 8b depict E2E delay, comprising transmission time (T_{tx}), propagation delay (T_{prop}), queuing delay (T_{queue}), and processing delay (T_{proc}), where T_{proc} includes cryptographic operations ($T_{sign} + T_{verify}$) [54]. Following IEEE 802.11p specifications, beacons are broadcast every 100 ms [46]. At $N = 10$, PQ-TDAA-Falcon-512 achieves 8.127 ms, comfortably below the 100 ms beacon-interval budget, while PQ-TDAA-Dilithium2 incurs 176.987 ms, exceeding the 100 ms beacon interval threshold.

The analysis of delay composition reveals that the cryptographic operation ($T_{sign} + T_{verify}$) contributes only 0.1194 ms (0.067% of total E2E delay) for Dilithium2, indicating that the remaining 176.8676 ms (99.93%) originates from network-layer overhead, primarily MAC queuing (T_{queue}) and transmission time (T_{tx}). This network-layer dominance is consistent with Bilstrup et al.'s findings that 802.11p broadcast performance degrades significantly with larger packet sizes due to increased channel occupancy and contention [55]. The delay penalty for Dilithium2 stems from its five-fragment beacon structure (6254 bytes) compared to Falcon-512's two-fragment structure (2331 bytes).

At high density ($N = 100$), both schemes exhibit increased delays (Dilithium2: 404.464 ms; Falcon-512: 253.273 ms) as the network approaches saturation. Table 21 quantifies Falcon dominance—that is, 8.13 ms vs. 176.99 ms ($N = 10$, $22\times$ faster) and 253 ms vs. 404 ms ($N = 100$, $1.6\times$ faster)—consistently meeting the ETSI 100 ms safety bound across operational densities ($N \leq 50$). Notably, authentication delays remain constant across all densities (Dilithium2: 119.3604 μ s; Falcon-512: 269.734 μ s), confirming that signature size, rather than cryptographic speed, is the critical performance bottleneck in VANETs.

5.9.4. Authentication (Auth) Delay Analysis

Authentication delay measures the time from when the sender possesses data until the receiver can authenticate the message [56]. In signature-based protocols, this delay

comprises both the signature generation time at the sender (T_{sign}) and the cryptographic verification time at the receiver (T_{verify}), representing the complete processing overhead required to authenticate a received beacon message. This metric is essential for evaluating the cryptographic overhead of various signature schemes and assessing their suitability for resource-constrained vehicular environments. The average authentication delay is computed as follows:

$$Average\ Auth\ Delay = \frac{\sum_{i=1}^N (T_{sign,i} + T_{verify,i})}{N} \quad (6)$$

The auth delay metric remained constant across all schemes, regardless of vehicle density, as shown in Table 21 and Figure 8c. Both PQ-TDAA signature variants demonstrated vastly superior computational efficiency. The implementation of Dilithium2 is faster, completing authentication in approximately 119.3604 μ s (66.9184 μ s for signing and 52.442 μ s for verification). Falcon-512, in comparison, takes approximately 269.734 μ s (190.9926 μ s for signing and 78.7414 μ s for verification), making PQ-TDAA-Dilithium2 roughly $2.3\times$ faster. This speed difference mainly stems from their designs: The Dilithium2 variant uses module lattices with simple polynomial operations, which modern CPUs handle very efficiently, whereas Falcon's NTRU-based structure requires floating-point computations during signing, which are inherently slower.

5.9.5. Goodput Analysis

Table 21 and Figure 8d report the receiver-side goodput, defined as the successfully delivered beacon payload per unit time [54], computed as:

$$Goodput\ [kbps] = \frac{packet\ received \times beacon\ size \times 8}{Simulation\ time \times 1000} \quad (7)$$

In sparse traffic ($N = 10$ vehicles), both PQ-TDAA instantiations already achieve highly efficient operation, with PQ-TDAA-Falcon-512 delivering strong goodput while keeping end-to-end delay well below the 100 ms safety bound. As vehicle density increases, goodput initially improves due to more frequent beacon exchanges, but eventually degrades under severe MAC-layer contention and fragmentation. Even in the most congested stress-test scenario ($N = 100$ vehicles), PQ-TDAA-Falcon-512 still attains 64,730.47 kbps of goodput, while the corresponding PDR of 3.56% reflects the expected impact of channel saturation rather than a limitation of the cryptographic layer. Falcon-512 achieves a $2.6\times$ goodput advantage at $N = 100$ (64.73 Mbps vs. 24.95 Mbps), confirming that signature compactness is the primary scalability driver in bandwidth-constrained VANETs. Overall, the comparison between Dilithium2 and Falcon-512 shows that, in bandwidth-constrained IEEE 802.11p VANETs, compact signatures and network-layer efficiency have a greater influence on end-to-end performance than raw cryptographic speed, since the cryptographic cost itself already remains well within practical real-time limits.

5.10. Discussion

The IP fragmentation protocol requires all fragments of a datagram to arrive successfully for reassembly [57]. If any fragment is lost, the entire datagram must be retransmitted. This all-or-nothing delivery creates exponential performance degradation under packet loss, similar to phenomena observed in wireless broadcast networks using IEEE 802.11p [58]. For a datagram fragmented into n pieces with per-fragment collision probability p , a simplified analytical model approximates the delivery success probability as

$$P_{success} = (1 - p)^n \quad (8)$$

which illustrates how a higher number of fragments amplifies the effective impact of channel impairments in dense beaconing scenarios [59].

Under IEEE 802.11p, the MTU is 1500 bytes [53,60], and PQ-TDAA beacon fragmentation differs significantly from PQ-TDAA-Dilithium2 (6254 bytes), which divides into five fragments, and PQ-TDAA-Falcon-512 (2331 bytes), which divides into two fragments. The corresponding delivery probability ratio between the two instantiations under the $(1 - p)^n$ model is:

$$Ratio = \frac{(1 - p)^2}{(1 - p)^5} = \frac{1}{(1 - p)^3} \quad (9)$$

which grows rapidly as p enters the moderate-to-high loss regime. Our NS-3 simulations report a $6.98 \times$ PDR gap at a vehicle density of $N = 100$, and, if interpreted through this simplified independent-fragment model, the gap corresponds to an effective per-fragment loss probability of approximately 0.55, typical of heavily congested 802.11p safety-beacon channels. This model is used only to provide qualitative intuition about the exponential penalty of fragmentation rather than to perform exact calibration, because real VANET fragment losses are driven by complex MAC-layer contention effects rather than purely independent per-fragment events. The analysis therefore reinforces a central design principle of PQ-TDAA that, in post-quantum VANETs, network performance depends far more on reducing signature size and the resulting number of fragments than on squeezing out small gains in signing speed.

In IEEE 802.11p VANETs, network scalability is governed not only by cryptographic computation latency but fundamentally by MAC-layer fragmentation and contention effects. Let S denote the total beacon size (payload, headers, and cryptographic material) and M the MTU; the number of required MAC-layer fragments scales as $k = \lceil \frac{S}{M} \rceil$. Under CSMA/CA, each fragment independently contends for channel access, so if the collision probability per fragment is p , the probability of successful beacon delivery scales approximately as $(1 - p)^k$. This exponential relationship implies that increases in signature size induce a non-linear collapse in PDR at high vehicle densities, even when cryptographic verification time is modest. This mechanism explains why PQ-TDAA instantiated with Falcon-512 achieves higher PDR and goodput than Dilithium2, despite higher per-signature computation latency: Falcon-512's compact signatures reduce fragmentation, lower effective collision probability, and preserve MAC-layer stability. Consequently, in bandwidth-constrained and contention-dominated VANET environments, signature compactness, not raw cryptographic speed, emerges as the dominant determinant of network-level performance, providing a theoretical foundation for the empirical NS-3 results.

The behavior observed in our NS-3 experiments is consistent with prior analyses of IEEE 802.11p broadcast performance, which report that the loss rate of WAVE safety packets increases markedly with packet size and vehicle density due to collisions, hidden terminals, and channel switching effects [59]. In our post-quantum setting, the larger Dilithium2-based beacons produce more fragments and therefore more extended channel occupancy, exacerbating these mechanisms and leading to pronounced degradation in PDR, E2E delay, and goodput compared with the more compact Falcon-512 instantiation. Existing performance characterizations of post-quantum signatures show that Falcon offers significantly smaller signatures at the cost of somewhat higher signing and verification time than Dilithium. Our PQ-TDAA results complement those studies by demonstrating that, in dense VANET beaconing scenarios, signature compactness and the resulting reduction in fragmentation are more critical for system-level performance than marginal differences in cryptographic speed, providing VANET-specific evidence to guide the selection of PQC algorithms for future V2X deployments.

6. Conclusions

The proposed PQ-TDAA protocol introduces a lattice-based, post-quantum secure design specifically engineered for the stringent constraints of VANET environments. Beyond algorithmic efficiency, the protocol is architected to be regulatory-compliant, aligning with ETSI TS 102 941 standards by managing the entire pseudonym lifecycle—from secure issuance and updates to efficient revocation. This is achieved through a distributed infrastructure model, where CA and RSU collaborate to manage vehicle identities without centralized bottlenecks.

To ensure high performance, NIST-standardized primitives are systematically integrated. The architecture utilizes a hybrid KEM-DEM construction, combining ML-KEM-512 with hardware-accelerated AES-256-GCM AEAD to provide real-time data confidentiality. To optimize bandwidth, PQ-TDAA leverages CRYSTALS-Dilithium2 and Falcon-512 digital signatures. By replacing the original ZKP with a simplified Schnorr-like proof via the Fiat–Shamir transform, the scheme reduces proof sizes by 69.2% compared to V-LDAA.

PQ-TDAA instantiations with Dilithium2 and Falcon-512 exhibit lightweight computation, completing all protocol phases in under 2 s. The Falcon-512 variant achieves a beacon size of 2331 bytes (62.73% smaller than Dilithium2), confirming signature compactness as the key performance driver. NS-3 analysis shows PQ-TDAA-Falcon-512 achieving 8.127 ms and 49.699 ms end-to-end delays at $N = 10/20$ vehicles, with goodput scaling to 64.73 Mbps—standard-compliant for ≤ 50 vehicle densities.

Real-world ARM validation on automotive-grade Cortex-A76 (Raspberry Pi 5) confirms that both variants achieve sub-0.5 ms complete V2V cycles (beacon generation + verification), well within 100 ms beacon intervals, bridging theoretical design with practical OBU deployment and validating real-time embedded suitability. Future work will explore PQ-TDAA standardization for 5G NR-V2X, develop NS-3/SUMO-coupled realistic traffic scenarios, and analyze side-channel countermeasures for production deployment.

Author Contributions: Conceptualization, E.R.A., K.R., R.H., T.S.G., M.S., A.A.L. and A.R.H.; data curation, K.R., R.H., T.S.G., M.S., A.A.L. and A.R.H.; formal analysis, E.R.A., K.R., R.H., T.S.G., M.S., A.A.L. and A.R.H.; funding acquisition, K.R. and M.S.; investigation, K.R., R.H., T.S.G., M.S., A.A.L. and A.R.H.; methodology, E.R.A., K.R., R.H., T.S.G., M.S., A.A.L. and A.R.H.; project administration, E.R.A.; resources, E.R.A., K.R. and M.S.; software, E.R.A., A.A.L. and A.R.H.; supervision, K.R., R.H., M.S. and T.S.G.; validation, K.R., R.H., T.S.G., M.S., A.A.L. and A.R.H.; visualization, E.R.A., A.A.L. and A.R.H.; writing—original draft, E.R.A.; writing—review and editing, E.R.A., K.R., R.H., T.S.G., M.S., A.A.L. and A.R.H. All authors have read and agreed to the published version of the manuscript.

Funding: This publication was supported by Universitas Indonesia through Hibah Publikasi Terindeks Internasional (PUTI) Q1 Kolaborasi Internasional Scheme under Contract: PKS-291/UN2.RST/HKP.05.00/2025. The work of E.R. was supported by the Indonesia Endowment Fund for Education or Lembaga Pengelola Dana Pendidikan (LPDP), Ministry of Finance of the Republic of Indonesia, under Contract: KEP1526/LPDP/LPDP.3/2022.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

The pseudocode (Algorithm A1, Algorithm A2, Algorithm A3, Algorithm A4 and Algorithm A5) of simplified Schnorr using Fiat–Shamir transform implementation in our

PQ-TDAA framework, especially in the Join, Create, and Network Registration phases, is as follows:

Algorithm A1: System Setup

Input: Security parameter $\lambda = 128$

Output: Public parameters PP, matrices (A, B), trapdoor T

```

1: Set parameters:
  - Ring dimension  $d \leftarrow 512$ 
  - Lattice dimensions  $(n, m) \leftarrow (1024, 1024)$ 
  - Modulus  $q \leftarrow 7933$ 
  - Gaussian width  $\sigma \leftarrow 232.0$ 
  - Norm bounds:  $\beta_r \leftarrow 2\sqrt{n}$ ,  $\beta_z \leftarrow 1.2\sigma\sqrt{n}$ ,  $\beta_s \leftarrow 1.2\sigma\sqrt{n}$ 
2: //Build CDT for Efficient Gaussian Sampling
3:  $T \leftarrow \lceil \text{tail} \cdot \sigma \rceil$  //  $T = 2320$ 
4: for  $k = 0$  to  $T$  do
5:    $\text{pmf}[k] \leftarrow \exp(-k^2/2\sigma^2)$  //Discrete Gaussian PDF
6: end for
7:  $Z \leftarrow 2 \cdot \text{sum}(\text{pmf}) - \text{pmf}(0)$  //Normalization constant
8: for  $k = 0$  to  $T$  do
9:    $\text{cdf}[k] \leftarrow \text{sum}(\text{pmf}[0:k])$  //Cumulative distribution
10: end for
11: //Generate Public Matrices
12: Generate a random matrix  $A \leftarrow Z_{-q}^{\wedge}(m \times n)$ 
13: Generate a random matrix  $B \leftarrow Z_{-q}^{\wedge}(m \times n)$ 
14: //Store Parameters
15:  $\text{PP} \leftarrow \{\lambda, d, n, m, q, \sigma, \beta_r, \beta_z, \beta_s, \text{cdf}, T\}$ 
16: return (PP, A, B)

```

Algorithm A2: Join Phase—Vehicle Side

Input: Matrix B, public parameters PP

Output: Commitment c, ciphertext ct, proof π_1

```

//Step 1: Choose Vehicle ID
1:  $\text{VID} \leftarrow \{0,1\}^{128}$  //Random vehicle identifier
2:  $\mu_1 \leftarrow H_{256}(\text{VID})$  //SHA3-256 hash
//Step 2: Sample Secret
3: repeat
4:    $r \leftarrow \text{Uniform}(\{-2, -1, 0, 1, 2\}^n)$ 
5: until  $\|r\| \leq \beta_r$ 
//Step 3: Compute Commitment
6:  $G_r \leftarrow \text{SHAKE}_{256}(r, 32)$ 
7:  $\text{seed}_h \leftarrow \text{SHAKE}_{256}(G_r \parallel \mu_1, 32)$ 
8:  $h \leftarrow \text{HashToVector}(\text{seed}_h, m)$  //Expand to  $Z_{-q}^{\wedge}m$ 
9:  $c \leftarrow B \cdot r + h \bmod q$ 
//Step 4: Encrypt ( $\mu_1, r$ )
10:  $\text{payload} \leftarrow \mu_1 \parallel \text{Encode}(r)$ 
11:  $\text{seed}_k \leftarrow \text{SHAKE}_{256}(\text{"EK"} \parallel c, 64)$ 
12:  $\text{nonce} \leftarrow \{0,1\}^{96}$ 
13:  $\text{keystream} \leftarrow \text{SHAKE}_{256}(\text{seed}_k \parallel \text{nonce}, |\text{payload}|)$ 
14:  $\text{ct\_data} \leftarrow \text{payload} \oplus \text{keystream}$ 
15:  $\text{tag} \leftarrow H_{256}(\text{"AAD"} \parallel \text{nonce} \parallel \text{ct\_data})[0:128]$ 
16:  $\text{ct} \leftarrow (\text{nonce}, \text{ct\_data}, \text{tag})$ 
//Step 5: Generate Zero-Knowledge Proof (ZKP)  $\pi_1$ 
17: attempt  $\leftarrow 0$ 
18: repeat
19:  $y \leftarrow D_{\sigma}n$  //Sample Gaussian masking
20:  $t_1 \leftarrow B \cdot y \bmod q$  //Commitment

```

Algorithm A2: Cont.

```

21: // Fiat-Shamir challenge
22: msg  $\leftarrow$  "JOIN" || B || c || ct || t1
23: ch  $\leftarrow$  {−1, +1} // H(msg) mod 2
24: z  $\leftarrow$  y + ch · r // Response
25: attempt  $\leftarrow$  attempt + 1
26: until ||z||  $\leq$   $\beta_z$  OR attempt > 200
27: if attempt > 200 then
28: return  $\perp$  // Proof generation failed
29: end if
30:  $\pi_1 \leftarrow$  (t1, z, ch)
31: return (c, ct,  $\pi_1$ )

```

Algorithm A3: Join Phase—CA Side

Input: Matrices (A, B), commitment c, ciphertext ct, proof π_1
Output: Credential s or $\perp$

```

// Parse proof
1: (t1, z, ch)  $\leftarrow$   $\pi_1$ 
2: (nonce, ct_data, tag)  $\leftarrow$  ct
// Step 1: Decrypt Ciphertext
3: seed_k  $\leftarrow$  SHAKE256("EK" || c, 64)
4: keystream  $\leftarrow$  SHAKE256(seed_k || nonce, |ct_data|)
5: payload  $\leftarrow$  ct_data  $\oplus$  keystream
6:  $\mu_1' \leftarrow$  payload [0:256]
7: r_bytes  $\leftarrow$  payload [256:]
8: r'  $\leftarrow$  Decode(payload [256:])
// Step 2: Norm Checks
9: if ||r'|| >  $\beta_r$  then return  $\perp$ 
10: if ||z|| >  $\beta_z$  then return  $\perp$ 
// Step 3: Verify Commitment
11: G_r'  $\leftarrow$  SHAKE256(r', 32)
12: seed_h  $\leftarrow$  SHAKE256(G_r' ||  $\mu_1'$ , 32)
13: h'  $\leftarrow$  HashToVector(seed_h, m)
14: c_check  $\leftarrow$  B · r' + h' mod q
15: if c_check  $\neq$  c then return  $\perp$ 
// Step 4: Verify Challenge
16: c_bytes  $\leftarrow$  Encode_Modq(c)
17: t1_bytes  $\leftarrow$  Encode_Modq(t1)
18: ct_bytes  $\leftarrow$  nonce || ct_data || tag
19: msg  $\leftarrow$  "JOIN" || c_bytes || ct_bytes || t1_bytes
20: ch_bytes  $\leftarrow$  H256(msg)
21: ch_expected  $\leftarrow$  +1 if ch_bytes [0] mod 2 = 0 else −1
22: if ch  $\neq$  ch_expected then return  $\perp$ 
// Step 5: Verify Zero-Knowledge Proof Equation
23: left  $\leftarrow$  B · z − ch · c + ch · h' mod q
24: if left  $\neq$  t1 then return  $\perp$ 
// Step 6: Issue Credential Using Simplified Gaussian Sampling
25: target_norm  $\leftarrow$  0.4 ·  $\beta_s$ 
26: repeat
27:   s_base  $\leftarrow$  D_σ^n // Sample Gaussian vector
28:   if ||s_base|| > 0 then
29:     scale  $\leftarrow$  target_norm / ||s_base||
30:     s  $\leftarrow$  Round(s_base · scale) // Scale to target norm
31:   else
32:     s  $\leftarrow$  D_σ^n // Retry if zero vector
33:   end if
34: until ||s||  $\leq$  1.2 · target_norm
35: return s // Credential successfully issued

```

Algorithm A4: Create Phase—OBU SideInput: Matrices (A, B), secrets (r, s), message μ_2 Output: Pseudonym ρ , proof π_2

```

//Step 1: Compute Pseudonym
1:  $\rho \leftarrow B \cdot r \bmod q$ 
//Step 2: Compute Auxiliary Data C
2:  $C \leftarrow A \cdot s \bmod q$ 
//Step 3: Compute Message Hash
3:  $\text{seed}_h \leftarrow \text{SHAKE}_{256}(\rho \parallel \mu_2, 32)$ 
4:  $h \leftarrow \text{HashToVector}(\text{seed}_h, m)$ 
//Step 4: Generate Zero-Knowledge Proof (ZKP)  $\pi_2$ 
5:  $\text{attempt} \leftarrow 0$ 
6: repeat
7:    $y_r \leftarrow D_{\sigma^n}$  //Sample Gaussian maskings
8:    $y_s \leftarrow D_{\sigma^n}$ 
9:    $t \leftarrow A \cdot y_s - B \cdot y_r \bmod q$  //Commitment
11:  //Fiat-Shamir challenge
12:   $\rho_{\text{bytes}} \leftarrow \text{Encode\_Modq}(\rho)$ 
13:   $C_{\text{bytes}} \leftarrow \text{Encode\_Modq}(C)$ 
14:   $t_{\text{bytes}} \leftarrow \text{Encode\_Modq}(t)$ 
15:   $\text{msg} \leftarrow \text{"CREATE"} \parallel \rho_{\text{bytes}} \parallel C_{\text{bytes}} \parallel t_{\text{bytes}} \parallel \mu_2$ 
16:   $\text{ch\_bytes} \leftarrow H_{256}(\text{msg})$ 
17:   $\text{ch} \leftarrow +1$  if  $\text{ch\_bytes}[0] \bmod 2 = 0$  else  $-1$  //Binary challenge
18:   $z_r \leftarrow y_r + \text{ch} \cdot r$  //Response for r
19:   $z_s \leftarrow y_s + \text{ch} \cdot s$  //Response for s
20:   $\text{attempt} \leftarrow \text{attempt} + 1$ 
21: until  $(\|z_r\| \leq \beta_z \text{ AND } \|z_s\| \leq \beta_s)$  OR  $\text{attempt} > 500$ 
22: if  $\text{attempt} > 500$  then
23:   return  $\perp$ 
24: end if
25:  $\pi_2 \leftarrow (t, z_r, z_s, \text{ch})$ 
26:  $\text{auxiliary} \leftarrow C$ 
27: return  $(\rho, \pi_2, C)$ 

```

Algorithm A5: Network Registration Phase—RSU VerificationInput: Matrices (A, B), pseudonym ρ , proof π_2 , message μ_2 , timestamp t_{now}

Output: Accept (1) or Reject (0)

```

//Parse proof
1:  $(\rho, \pi_2) \leftarrow \text{signature}$ 
2:  $(t, z_r, z_s, \text{ch}) \leftarrow \pi_2$ 
3:  $C \leftarrow \text{auxiliary}$ 
4:  //Step 2: Norm Checks
5: if  $\|z_r\| > \beta_z$  then return 0
6: if  $\|z_s\| > \beta_s$  then return 0
//Step 3: Verify Challenge
7:  $\rho_{\text{bytes}} \leftarrow \text{Encode\_Modq}(\rho)$ 
8:  $C_{\text{bytes}} \leftarrow \text{Encode\_Modq}(C)$ 
9:  $t_{\text{bytes}} \leftarrow \text{Encode\_Modq}(t)$ 
10:  $\text{msg} \leftarrow \text{"CREATE"} \parallel \rho_{\text{bytes}} \parallel C_{\text{bytes}} \parallel t_{\text{bytes}} \parallel \mu_2$ 
11:  $\text{ch\_bytes} \leftarrow H_{256}(\text{msg})$ 
12:  $\text{ch\_expected} \leftarrow +1$  if  $\text{ch\_bytes}[0] \bmod 2 = 0$  else  $-1$ 
13: if  $\text{ch} \neq \text{ch\_expected}$  then return 0
//Step 4: Verify Modified Proof Equation
//Verification check:  $A \cdot z_s - B \cdot z_r =? t + \text{ch} \cdot (C - \rho)$ 

```

Algorithm A5: Cont.

```

14: LHS  $\leftarrow$  A.z_s - B.z_r mod q //Compute left hand side
15: RHS  $\leftarrow$  t + ch.(C- $\rho$ ) //Compute right hand side
16: if LHS  $\neq$  RHS then
17:   return 0 //Verification failed
18: end if
19: return 1 //Verification successful

```

References

- Manasrah, A.; Yaseen, Q.; Al-Aqrabi, H.; Liu, L. Identity-Based Authentication in VANETs: A Review. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 4260–4282. [\[CrossRef\]](#)
- Jan, S.A.; Amin, N.U.; Othman, M.; Ali, M.; Umar, A.I.; Basir, A. A Survey on Privacy-Preserving Authentication Schemes in VANETs: Attacks, Challenges and Open Issues. *IEEE Access* **2021**, *9*, 153701–153726. [\[CrossRef\]](#)
- Agustina, E.R.; Ramli, K.; Hakim, A.R.; Harwahu, R. A Systematic Literature Review on Privacy Preservation in VANETs: Trends, Challenges, and Future Directions. *IEEE Access* **2025**, *13*, 88421–88444. [\[CrossRef\]](#)
- Azam, F.; Yadav, S.K.; Priyadarshi, N.; Padmanaban, S.; Bansal, R.C. A Comprehensive Review of Authentication Schemes in Vehicular Ad-Hoc Network. *IEEE Access* **2021**, *9*, 31309–31321. [\[CrossRef\]](#)
- Raya, M.; Papadimitratos, P.; Hubaux, J.P. Securing Vehicular Communications. *IEEE Wirel. Commun.* **2006**, *13*, 8–15. [\[CrossRef\]](#)
- Brickell, E.; Camenisch, J.; Chen, L. Direct anonymous attestation. In Proceedings of the 11th ACM Conference on Computer and Communications Security, Washington, DC, USA, 25–29 October 2004; pp. 132–145.
- Shim, K.A. A Survey on Post-Quantum Public-Key Signature Schemes for Secure Vehicular Communications. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 14025–14042. [\[CrossRef\]](#)
- Chen, L.; Tu, T.; Yu, K.; Zhao, M.; Wang, Y. V-LDAA: A New Lattice-Based Direct Anonymous Attestation Scheme for VANETs System. *Secur. Commun. Netw.* **2021**, 4660875. [\[CrossRef\]](#)
- Fiat, A.; Shamir, A. How to prove yourself: Practical solutions to identification and signature problems. In *Proceedings of the Conference on the Theory and Application of Cryptographic Techniques, Linköping, Sweden, 20–22 May 1986*; Springer: Berlin/Heidelberg, Germany, 1986; pp. 186–194.
- Schnorr, C.P. Efficient signature generation by smart cards. *J. Cryptol.* **1991**, *4*, 161–174. [\[CrossRef\]](#)
- Lyubashevsky, V. Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures. In *Proceedings of the Advances in Cryptology—ASIACRYPT 2009, Tokyo, Japan, 6–10 December 2009*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 598–616.
- Lyubashevsky, V. Lattice Signatures without Trapdoors. In *Proceedings of the Advances in Cryptology—EUROCRYPT 2012, Cambridge, UK, 15–19 April 2012*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 738–755.
- Beullens, W.; Lyubashevsky, V.; Nguyen, N.K.; Seiler, G. *Lattice-Based Blind Signatures: Short, Efficient, and Round-Optimal*; Association for Computing Machinery: New York, NY, USA, 2023; pp. 16–29.
- ETSI TS 102 941; Intelligent Transport Systems (ITS); Security; Trust and Privacy Management; Release 2. ETSI: Sophia Antipolis, France, 2022.
- National Institute of Standards and Technology. *FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2024.
- National Institute of Standards and Technology. *FIPS 204 Module-Lattice-Based Digital Signature Standard*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2024.
- Fouque, P.-A.; Gajland, P.; de Groote, H.; Janneck, J.; Kiltz, E. A Closer Look at Falcon. *Cryptology ePrint Archive*, Paper 2024/1769. 2024. Available online: <https://eprint.iacr.org/2024/1769> (accessed on 16 February 2026).
- Open Quantum Safe Project. Falcon. Available online: <https://openquantumsafe.org/liboqs/algorithms/sig/falcon.html> (accessed on 3 November 2025).
- Prajapat, S.; Gautam, D.; Kumar, P.; Jangirala, S.; Das, A.K.; Park, Y.; Lorenz, P. Secure Lattice-Based Aggregate Signature Scheme for Vehicular Ad Hoc Networks. *IEEE Trans. Veh. Technol.* **2024**, *73*, 12370–12384. [\[CrossRef\]](#)
- Han, L.; Cao, S.; Yang, X.; Zhang, Z. Privacy Protection of VANET Based on Traceable Ring Signature on Ideal Lattice. *IEEE Access* **2020**, *8*, 206581–206591. [\[CrossRef\]](#)
- Wani, N.M.; Verma, G.K.; Chamola, V. Dynamic Anonymous Quantum-Secure Batch-Verifiable Authentication Scheme for VANET. *IEEE Trans. Consum. Electron.* **2024**, *70*, 7112–7120. [\[CrossRef\]](#)
- Li, L.; Hsu, C.; Au, M.H.; Cui, J.; Harn, L.; Zhao, Z. Lattice-Based Conditional Privacy-Preserving Batch Authentication Protocol for Fog-Assisted Vehicular Ad Hoc Networks. *IEEE Trans. Inf. Forensics Secur.* **2024**, *19*, 9629–9642. [\[CrossRef\]](#)
- Li, Q.; He, D.; Yang, Z.; Xie, Q.; Choo, K.K.R. Lattice-Based Conditional Privacy-Preserving Authentication Protocol for the Vehicular Ad Hoc Network. *IEEE Trans. Veh. Technol.* **2022**, *71*, 4336–4347. [\[CrossRef\]](#)

24. Liu, G.; Li, H.; Le, J.; Wang, N.; Liu, Y.; Xiang, T. LRPCA: Lattice-Based Robust and Conditional Privacy-Preserving Authentication for VANETs. *IEEE Trans. Veh. Technol.* **2024**, *74*, 4698–4712. [[CrossRef](#)]
25. Xu, S.W.; Yu, S.H.; Bai, Y.J.; Yue, Z.Y.; Liu, Y.L. LB-CLAS: Lattice-based conditional privacy-preserving certificateless aggregate signature scheme for VANET. *Veh. Commun.* **2024**, *50*, 100843. [[CrossRef](#)]
26. Arm Limited. *Cortex-A76AE Automotive Enhanced: Datasheet*; Arm Limited: Cambridge, UK, 2024.
27. Raspberry Pi Ltd. *Raspberry Pi 5 Product Brief*; Raspberry Pi Ltd.: Cambridge, UK, 2025.
28. Kannwischer, M.J.; Rijneveld, J.; Schwabe, P.; Stoffelen, K. pqm4: Testing and Benchmarking NIST PQC on ARM Cortex-M4; IACR Cryptology ePrint Archive, Paper 2019/844. 2019. Available online: <https://repository.uibn.ru.nl/bitstream/handle/2066/210214/210214.pdf> (accessed on 16 February 2026).
29. John, L.K. *Performance Evaluation Techniques, Tools and Benchmarks*; University of Texas at Austin: Austin, TX, USA, 2002.
30. Fog, A. *The Microarchitecture of Intel, AMD and VIA CPUs: An Optimization Guide for Assembly Programmers and Compiler Makers*; Technical University of Denmark: Kongens Lyngby, Denmark, 2021.
31. La Manna, M.; Trecozzi, L.; Perazzo, P.; Saponara, S.; Dini, G. Performance Evaluation of Attribute-Based Encryption in Automotive Embedded Platform for Secure Software Over-The-Air Update. *Sensors* **2021**, *21*, 515. [[CrossRef](#)]
32. Lyubashevsky, V.; Nguyen, N.K.; Plançon, M. *Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General*; Springer Nature Switzerland: Cham, Switzerland, 2022.
33. National Institute of Standards and Technology. *Advanced Encryption Standard (AES); 197 (FIPS 197)*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2023.
34. Dworkin, M. *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC; 800-38D*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2007.
35. Whitefield, J.; Chen, L.; Giannetos, T.; Schneider, S.; Treharne, H. Privacy-enhanced capabilities for VANETs using direct anonymous attestation. In Proceedings of the 2017 IEEE Vehicular Networking Conference (VNC), Torino, Turin, Italy, 27–29 November 2017; pp. 123–130.
36. Chen, L.; Ng, S.L.; Wang, G. Threshold Anonymous Announcement in VANETs. *IEEE J. Sel. Areas Commun.* **2011**, *29*, 605–615. [[CrossRef](#)]
37. Desmoulins, N.; Diop, A.; Rafflé, Y.; Traoré, J.; Gratesac, J. Practical Anonymous Attestation-based Pseudonym Schemes for Vehicular Networks. In Proceedings of the 2019 IEEE Vehicular Networking Conference (VNC), Los Angeles, CA, USA, 4–6 December 2019; pp. 1–8.
38. Dharminder, D.; Mishra, D. LCPA: Lattice-based conditional privacy preserving authentication in vehicular communication. *Trans. Emerg. Telecommun. Technol.* **2020**, *31*, e3810. [[CrossRef](#)]
39. Wen, J.; Bai, L.; Yang, Z.; Zhang, H.; Wang, H.; He, D. LaRRS: Lattice-Based Revocable Ring Signature and Its Application for VANETs. *IEEE Trans. Veh. Technol.* **2024**, *73*, 739–753. [[CrossRef](#)]
40. Mundhe, P.; Yadav, V.K.; Verma, S.; Venkatesan, S. Efficient Lattice-Based Ring Signature for Message Authentication in VANETs. *IEEE Syst. J.* **2020**, *14*, 5463–5474. [[CrossRef](#)]
41. Wang, F.; Gu, M.; Xiao, H.; Wang, Y.; Cao, C. Secure and Efficient Attribute-based Batch Authentication Scheme for VANETs. *IEEE Netw.* **2025**. [[CrossRef](#)]
42. Agustina, E.R.; Hakim, A.R.; Ramli, K. Modeling Data Security and Privacy Threats for VANET using STRIDE and LINDDUN. In Proceedings of the 2024 2nd International Conference on Software Engineering and Information Technology (ICoSEIT), Bandung, Indonesia, 28–29 February 2024; pp. 114–119.
43. Cremers, C.J.F. The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols. In *Proceedings of the Computer Aided Verification, Princeton, NJ, USA, 7–14 July 2008*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 414–418.
44. Chen, Y.; Nguyen, P.Q. BKZ 2.0: Better Lattice Security Estimates. *Lect. Notes Comput. Sci.* **2011**, *7073*, 1–20. [[CrossRef](#)]
45. Alagic, G.; Apon, D.; Cooper, D.; Dang, Q.; Kelsey, J.; Liu, Y.-K.; Miller, C.; Moody, D.; Peralta, R.; Perlner, R.; et al. *Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2022.
46. J2945_1_201603; On-Board System Requirements for V2V Safety Communications. SAE International: Warrendale, PA, USA, 2016.
47. Kim, Y.; Song, J.; Youn, T.-Y.; Seo, S.C. Crystals-Dilithium on ARMv8. *Secur. Commun. Netw.* **2022**, *2022*, 5226390. [[CrossRef](#)]
48. Kim, Y.; Song, J.; Seo, S.C. Accelerating Falcon on ARMv8. *IEEE Access* **2022**, *10*, 44446–44460. [[CrossRef](#)]
49. Kementerian Perhubungan Republik Indonesia. *Peraturan Menteri Perhubungan Nomor 96 Tahun 2015 Tentang Pedoman Pelaksanaan Kegiatan Manajemen dan Rekayasa Lalu Lintas*; Kementerian Perhubungan Republik Indonesia: Jakarta, Indonesia, 2015.
50. Karnadi, F.K.; Mo, Z.H.; Lan, K.-C. Rapid Generation of Realistic Mobility Models for VANET. In Proceedings of the IEEE Wireless Communications and Networking Conference, Hong Kong, China, 11–15 March 2007; pp. 2508–2513.
51. Sommer, C.; German, R.; Dressler, F. Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. *IEEE Trans. Mob. Comput.* **2011**, *10*, 3–15. [[CrossRef](#)]

52. Ma, X.; Chen, X. Performance Analysis of IEEE 802.11 Broadcast Scheme in Ad Hoc Wireless LANs. *IEEE Trans. Veh. Technol.* **2008**, *57*, 3757–3768. [[CrossRef](#)]
53. Qiu, H.J.F.; Ho, I.W.-H.; Tse, C.K.; Xie, Y. A Methodology for Studying 802.11p VANET Broadcasting Performance with Practical Vehicle Distribution. *IEEE Trans. Veh. Technol.* **2016**, *65*, 8756–8769. [[CrossRef](#)]
54. Hota, L.; Nayak, B.P.; Kumar, A.; Sahoo, B.; Ali, G.G.M.N. A Performance Analysis of VANETs Propagation Models and Routing Protocols. *Sustainability* **2022**, *14*, 1379. [[CrossRef](#)]
55. Bilstrup, K.; Uhlemann, E.; Ström, E.G.; Bilstrup, U. Evaluation of the IEEE 802.11p MAC Method for Vehicle-to-Vehicle Communication. In Proceedings of the IEEE 68th Vehicular Technology Conference (VTC Fall), Calgary, AB, Canada, 21–24 September 2008; pp. 1–5.
56. Studer, A.; Bai, F.; Bellur, B.; Perrig, A. Flexible, Extensible, and Efficient VANET Authentication. In *Proceedings of the 6th Embedded Security in Cars Conference; KICS: Hamburg, Germany*, 2008.
57. Postel, J. Internet Protocol; RFC 791; IETF. 1981. Available online: <https://www.rfc-editor.org/rfc/rfc791> (accessed on 20 December 2025).
58. Van Eenennaam, M.; Remke, A.; Heijenk, G. An Analytical Model for Beaconing in VANETs. In Proceedings of the IEEE Vehicular Networking Conference, Seoul, Republic of Korea, 14–16 November 2012; pp. 9–16.
59. Gonzalez, S.; Ramos, V. A Simulation-Based Analysis of the Loss Process of Broadcast Packets in WAVE Vehicular Networks. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 7430728. [[CrossRef](#)]
60. IEEE. *IEEE Standard for Information Technology—Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks—Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments*; IEEE: Piscataway, NJ, USA, 2010; pp. 1–51. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.