

# QR-LBTS: Quantum-Resilient Lattice-Based Threshold Signatures for Efficient Post-Quantum VANETs

Hao Yuan, *Student Member, IEEE*, Yuling Chen, Jiachun Li, *Member, IEEE*, Jing Sun, Minyi Guo, *Fellow, IEEE*

**Abstract**—Vehicular ad-hoc networks (VANETs) require secure and low-latency authentication to support reliable message exchange in dynamic vehicular environments. Although lattice-based threshold signatures provide post-quantum security and fault tolerance, existing schemes often suffer from high computational and communication overhead, which limits their practical deployment in VANETs. To address this issue, this paper proposes a *Quantum-Resilient Lattice-Based Threshold Signature* (QR-LBTS) framework for efficient post-quantum VANET authentication. Built upon Crystals-Dilithium and the RLWE assumption, QR-LBTS is designed to improve authentication efficiency, scalability, and robustness while preserving strong post-quantum security guarantees. Security is further analyzed under a unified security model against a quantum polynomial-time adversary, based on the RLWE assumption and the statistical indistinguishability of discrete Gaussian sampling. Security analysis shows that the proposed framework can resist data injection, man-in-the-middle, data leakage, and side-channel attacks. Experimental results further demonstrate that QR-LBTS achieves online signing latency of 0.35 s in the 2-party setting and 3.91 s in the 12-party setting, outperforming existing schemes and showing its practicality for threshold-based, security-critical VANET authentication tasks.

**Index Terms**—VANETs, Post-Quantum Cryptography, Lattice-Based Threshold Signature, Distributed Rejection Sampling.

## I. INTRODUCTION

WITH the emerging development of autonomous driving techniques, vehicular ad-hoc networks (VANETs) face more frequent and complex message sharing. As shown in Fig. 1, within VANET communication scenarios, frequent,

This research was supported by the Foundation of Key Points of the Joint Foundation of National Natural Science Foundation of China (U24A20241), the National Natural Science Foundation of China (62202118), the Major Scientific and Technological Special Project of Guizhou Province ([2024]014), the Scientific and Technological Research Projects from Guizhou Education Department (Qian jiao ji [2023]003), the Provincial Department of Science and Technology's Hundred level Innovation Talents Project (Guizhou Science and Technology Cooperation Platform Talents-GCC [2023]018), the Guizhou Province Major Project (Qiankehe Major Project No. [2024]003), and the Guizhou Provincial Innovation Talent Team for Cybersecurity Protection of Computing Power (Guizhou Science and Technology Cooperation Talents-CXTD[2025]029), Scientific and Technological Innovation Platform Research Project of Guizhou Province (CXPTXM[2025]024), Guizhou Provincial Basic Research Program (Natural Science) (Qiankehe Foundation MS[2026]207). (Corresponding author: Yuling Chen and Minyi Guo.)

Hao Yuan is with the State Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China (e-mail: gs.hyuan25@gzu.edu.cn).

Yuling Chen is with the State Key Laboratory of Public Big Data and the College of Computer Science and Technology, Guizhou University, Guiyang 550025, China (e-mail: ylchen3@gzu.edu.cn).

Jiachun Li is with the State Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China (e-mail: lijc@gzu.edu.cn).

Jing Sun is with the State Key Laboratory of Public Big Data, Guizhou University, Guiyang, 550025, China (e-mail: gs.jsun24@gzu.edu.cn).

Minyi Guo is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China. (e-mail: myguo@sjtu.edu.cn).

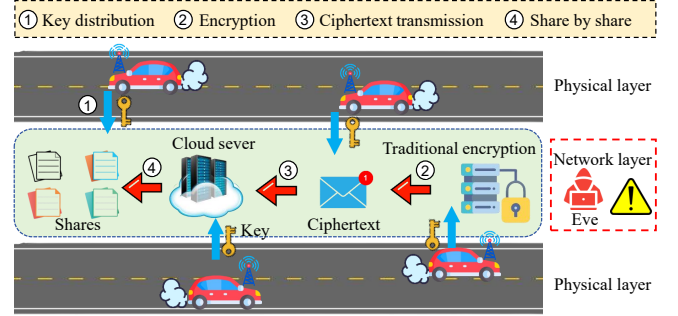


Fig. 1. Threshold signature scheme in traditional VANETs.

low-latency, and trustworthy message exchanges are indispensable among vehicles, roadside infrastructures, and cloud coordination platforms. Thus, it calls for a higher requirement for communication with stringent security and low-latency authentication [1]–[3].

Recent studies [4], [5] have shown that cyber attacks, for example, identity spoofing, unauthorized message injection, and replay attacks, can cause severe damage in message sharing of VANETs, which may compromise the integrity, confidentiality, and availability of vehicular communications. Specifically, such attacks can easily lead to severe efficiency issues (e.g., traffic congestion, unexpected accidents) or even critical security threats (e.g., privacy leakage, system crashes, financial loss). Therefore, ensuring reliable and timely authentication has become a fundamental requirement for the reliable operation of VANETs.

Threshold signatures are a promising candidate authentication scheme for VANETs, offering advantages over traditional encryption technologies [6]–[8], including low-latency authentication, scalability, and robustness against malicious attacks. Among these schemes, lattice-based cryptographic constructions have attracted significant attention as post-quantum solutions, providing inherent resistance to quantum attacks and broad applicability in areas such as quantum encryption [9], key exchange [10], and digital signatures [11]. In particular, lattice-based threshold signature technologies have emerged as a key research focus due to their combination of efficiency and strong security in multi-party computation scenarios, making them well-suited for the high communication demands of VANETs.

Despite these advantages, practical deployment in highly dynamic VANET environments remains challenging, especially when both strong security and timely authentication must be achieved simultaneously. In this work, we focus on the authentication layer of VANETs rather than on a generic

static distributed system. The VANET-specific challenges considered here arise mainly from three aspects: stringent latency requirements for safety-relevant message validation, rapidly changing participant availability due to vehicle mobility and dynamic topology, and increased communication overhead in large-scale or wide-area vehicular environments. These characteristics motivate the use of adaptive threshold verification and localized aggregation in QR-LBTS, so that the protocol can better accommodate dynamic participation and communication constraints in vehicular settings.

However, existing lattice-based threshold signature schemes still face two major challenges in practical deployment: 1) For security considerations, temporary shared keys and multi-party interaction phases inherently introduce potential risks, including unauthorized data injection, man-in-the-middle attacks, temporary secret leakage attack and side-channel attack. Unauthorized data injection may allow adversaries to forge traffic-related messages or malicious signature shares, thereby misleading cooperative decision-making among vehicles and roadside infrastructures. Man-in-the-middle attacks may occur during V2V or V2I message forwarding, where adversaries intercept, alter, or relay protocol messages to disrupt authentication correctness. Temporary secret leakage attacks are particularly dangerous during short-lived signing, share-exchange, or key-refresh phases, where partial exposure of ephemeral secrets may be exploited to infer signing information or undermine protocol integrity. Side-channel attacks may further arise from timing, power, or implementation-level leakage on resource-constrained vehicular devices, creating additional opportunities for adversaries to recover sensitive intermediate information. These threats are particularly challenging to detect and mitigate in large-scale, highly dynamic vehicular networks where vehicles frequently join and leave the network. Thus, it is crucial to overcome the aforementioned security threats in post-quantum VANETs. 2) For the efficiency part, traditional threshold signature schemes face significant challenges in computational efficiency and scalability. As the number of vehicles increases, the computational and communication costs of key generation, signing, and verification may grow substantially [12]–[14], resulting in non-negligible overhead. In large-scale distributed vehicular networks, multi-round interactions and complex multi-party computations further exacerbate latency, limiting real-time responsiveness. In wide-area VANETs, the aforementioned increasing overhead can significantly hinder the deployment of traditional schemes [15], [16], thereby challenging the efficiency and usability of VANETs.

To address these challenges, this paper proposes a Quantum-Resilient Lattice-Based Threshold Signatures (QR-LBTS) framework for multi-party VANETs. Built upon Crystals-Dilithium under the RLWE assumption, QR-LBTS aims to improve the efficiency, scalability, and post-quantum security of multi-party vehicular authentication. Unlike a generic threshold-signature deployment on static nodes, QR-LBTS is tailored to dynamic VANET environments by coupling adaptive threshold verification with Local Neighbor Weighted Computation (NWC), thereby tolerating fluctuating participant availability and reducing online communication rounds during

multi-party authentication. QR-LBTS is intended for security-critical threshold authentication tasks, such as RSU-assisted group authentication, platoon-level authorization, cooperative event validation, and infrastructure-mediated trust establishment, rather than replacing lightweight per-packet authentication for every high-frequency safety beacon. The main contributions of this work are summarized as follows:

- 1) **Adaptive threshold configuration.** We introduce an adaptive verification threshold mechanism that allows rapid validation once at least  $t_v$  participants approve, instead of requiring all nodes to engage. When fewer than  $t_v$  approvals are obtained, a complete verification involving at least  $t_s$  participants is invoked. This two-tier verification design significantly improves efficiency and flexibility under dynamic VANET conditions while preserving the original  $t_s$ -out-of- $n$  security guarantees of the underlying threshold signature scheme.
- 2) **Efficient local computation and aggregation.** We propose a Neighbor Weighted Consensus (NWC) for local weighted computation and effective merging, which supports the online and offline computation and aggregation. This method reduces the number of online communication rounds, lowering latency and communication overhead, and maintaining scalability even in large VANETs with multiple vehicles.
- 3) **Enhanced security.** To resist the potential attacks (e.g., data injection, man-in-the-middle, data leakage, and side-channel attacks), QR-LBTS strengthens computational uniformity and randomness throughout our designed protocol. In this protocol, we exploit rejection sampling, signature generation, Number Theoretic Transform (NTT), followed by a novel shared refresh mechanism to realize the security goal in VANETs.

According to extensive experiments and security analysis, our QR-LBTS can defend against data injection, man-in-the-middle, data leakage, and side-channel attacks, which can also significantly outperform existing lattice-based threshold signature schemes [17], [18] in both computation and communication efficiency. Specifically, with 12 participants, the overall overhead of QR-LBTS reaches a total of 13.8 s across all the online key generation and signing phases. The online signing phase requires 3.91 s, achieving about a  $4.45\times$  speedup over Tang et al. [17]. In addition, the key-generation phase requires only 3.39 s, corresponding to approximately 0.77 times the key-generation time of Tang et al. [17], highlighting the scheme's suitability for threshold-based, security-critical multi-party authentication tasks in large-scale distributed vehicular systems.

The structure of this paper is as follows: Section II reviews recent progress in threshold signature technology for VANETs, Section III introduces the basics of threshold signatures, Section IV provides a detailed description of our QR-LBTS, including algorithm design, protocol flow and security analysis, Section V experimentally validates the performance of our scheme, followed by the conclusion in Section VI.

## II. RELATED WORK

This section reviews recent progress in threshold signature technology for VANETs, covering four aspects: traditional threshold signature approaches in VANETs, single-party and multi-party attribute-based approaches in VANETs, post-quantum designs in VANETs and comparison of existing schemes.

### A. Traditional Threshold Signature Approaches in VANETs

Firstly, we introduce the traditional threshold signature approaches in VANETs. For example, Shamir *et al.* [19] proposed the pioneering secret sharing scheme, forming the cornerstone for subsequent threshold signature constructions, but it incurs high communication overhead and lacks practical optimization for distributed VANET environments. Boldyreva *et al.* [20] introduced an elliptic curve-based threshold signature, significantly improving computational efficiency and practical security, yet its reliance on classical hardness assumptions limits long-term resilience against quantum attacks. In VANETs, such classical schemes have been applied to authenticate cooperative driving messages and distributed control signals [21], ensuring low-latency verification and fault-tolerant message delivery [22]. Nevertheless, in high-mobility vehicular environments, these schemes remain constrained in terms of efficiency, scalability, and long-term security, especially under potential quantum attacks, highlighting the urgent need for lightweight, post-quantum secure threshold signature solutions.

### B. Single-party and Multi-party Attribute-Based Approaches in VANETs

In the context of VANETs, existing research on threshold signatures has extended toward attribute-based and privacy-preserving designs to address heterogeneous trust levels and fine-grained message verification. For single-party attribute schemes, Zhao *et al.* [23] proposed a Traceable and Forward-Secure Attribute-Based Signature (TFS-ABS) framework that derives signing keys from attributes, offering enhanced privacy guarantees but can introduce computational overhead in dense scenarios. Susilo *et al.* [24] developed a threshold attribute-based anonymous credential system for identity tracing and revocation, supporting selective verification in distributed environments, though frequent key updates and revocation management incur performance costs. Collectively, these works facilitate privacy-preserving authentication, collaborative management, and user anonymity [25], yet they still may face scalability challenges in large-scale, high-mobility VANETs.

For multi-party attribute schemes, lattice-based designs such as those by Shi *et al.* [26] and hierarchical identity-based variants [27] further improve forward security, dynamic user management, and post-quantum resistance. However, verification latency remains a critical bottleneck when the number of participating vehicles grows, limiting real-time applicability in cooperative driving scenarios. These efforts collectively call for more advanced attribute-based threshold signatures for VANETs, addressing both privacy and security, while requiring

the balance of efficiency, scalability, and robustness under high-mobility conditions.

### C. Post-Quantum Designs in VANETs

To address the threats posed by quantum computing in future autonomous transportation, post-quantum constructions have been proposed, including lattice-based linkable ring signatures [28]–[30] and RLWE-based multi-party signatures [31], providing quantum-resistant solutions for VANETs. Tang *et al.* [17] further introduced a lattice-based  $t$ -out-of- $n$  threshold signature scheme supporting distributed key generation, dynamic assignment, and periodic key refreshing, making it suitable for secure VANETs message authentication. Additionally, complementary works on forward-secure lattice protocols [32]–[34] and post-quantum attribute threshold designs [35]–[37] enhance robustness. Despite these advances in security and quantum resilience, large-scale applicability and efficient deployment remain significant challenges in high-mobility vehicular networks.

In summary, secure and efficient threshold signature schemes in VANETs are crucial to guarantee real-time message authentication, collaborative driving, and fault-tolerant coordination. Despite progress in efficiency, scalability, and robustness, existing works still struggle to reconcile strong quantum resistance with lightweight performance in large-scale deployments. To address these gaps, our QR-LBTS framework integrates RLWE-based key generation, MAC-enabled message sharing, forward-secure key refreshing, and distributed signing, offering a practical route to secure and scalable post-quantum threshold signatures in VANETs.

### D. Comparison of Existing Schemes

To provide a systematic overview of foundational and classical threshold signature schemes in VANETs, we summarize representative works in Tables I and II, which highlight their efficiency, functionality, and security trade-offs across [17] in Section A, [19] in Section B, and [23], [29], [30] in Section C. Shamir *et al.* [19] achieve low complexity and latency, but lack scalability and identity-based features, leaving them unsuitable for dynamic large-scale environments. Tang *et al.* [17] and Zhao *et al.* [23] provide partial functional support but suffer from higher overhead. Wen *et al.* [29] and Xie *et al.* [30] achieve a better balance between efficiency and scalability, though identity support is still absent. In terms of security, as shown in Table II, existing works can resist some threats such as data injection or man-in-the-middle attacks, but most remain vulnerable to temporary secret leakage or side-channel attacks. In contrast, as shown in Table I, our scheme achieves low complexity and latency while supporting both scalability and identity-based authentication, and further ensures comprehensive protection against all considered attacks.

## III. PRELIMINARIES

In this section, we introduce the basic notations, lattice-based cryptography foundations, and key building blocks that are essential for understanding the proposed QR-LBTS scheme.

TABLE I  
EFFICIENCY AND FUNCTIONALITY COMPARISON OF EXISTING WORKS.

| Existing work      | Efficiency     |             | Function    |                |
|--------------------|----------------|-------------|-------------|----------------|
|                    | Low complexity | Low latency | Scalability | Identity-based |
| Tang et al. [17]   | ✗              | ✗           | ✓           | ✓              |
| Shamir et al. [19] | ✓              | ✓           | ✗           | ✗              |
| Zhao et al. [23]   | ✗              | ✓           | ✗           | ✓              |
| Wen et al. [29]    | ✓              | ✓           | ✓           | ✗              |
| Xie et al. [30]    | ✓              | ✓           | ✓           | ✗              |
| <b>Our scheme</b>  | ✓              | ✓           | ✓           | ✓              |

TABLE II  
SECURITY COMPARISON OF EXISTING WORKS.

| Existing work      | Unauthorized data injection | Man-in-the-middle attack | Temporary secret leakage attack | Side-channel attack |
|--------------------|-----------------------------|--------------------------|---------------------------------|---------------------|
| Tang et al. [17]   | ✓                           | ✓                        | ✓                               | ✗                   |
| Shamir et al. [19] | ✗                           | ✗                        | ✗                               | ✗                   |
| Zhao et al. [23]   | ✓                           | ✗                        | ✓                               | ✗                   |
| Wen et al. [29]    | ✓                           | ✓                        | ✗                               | ✗                   |
| Xie et al. [30]    | ✓                           | ✓                        | ✓                               | ✗                   |
| <b>Our scheme</b>  | ✓                           | ✓                        | ✓                               | ✓                   |

### A. Notations

We denote  $n$  as the polynomial dimension and  $q$  as the prime modulus for operations over the ring  $\mathbb{Z}_q[X]/(X^n + 1)$ , with  $\eta$  controlling the standard deviation in discrete Gaussian sampling [38]. For each participant  $P_i$ ,  $s_i$  and  $t_i$  represent the private and public keys,  $\alpha_i$  is an auxiliary random polynomial,  $\Upsilon(t)(\xi)_i$  and  $\alpha(\xi)_i$  denote shared MAC values and key shares, and  $z_i$  and  $c_i$  denote partial signature and challenge bits. Random values  $r(j)$  are used for MAC aggregation, while  $\lambda_i$  represents the Lagrange coefficient for combining partial signatures. Continuous samples  $sa_i$  are discretized into  $ds_i$  via rejection sampling. All vectors are column vectors, matrices are bold uppercase letters, and repeated modular operations and multi-round computations in MAC Message Sharing and Shared Refresh enhance resistance against tampering and leakage attacks.

### B. Lattice Basics

In  $\mathbb{R}^n$  [39], a lattice  $\mathcal{L}$  is defined as the set of all integer linear combinations of basis vectors  $\mathbf{b}_1, \dots, \mathbf{b}_n$ :  $\mathcal{L} = \{\sum_{i=1}^n c_i \mathbf{b}_i \mid c_i \in \mathbb{Z}\}$ . To construct an efficient quantum-safe signature, we adopt the RLWE assumption, where in the polynomial ring  $R_q = \mathbb{Z}_q[X]/(X^N + 1)$ , a public polynomial  $a \in R_q$ , a secret polynomial  $s \in R_q$ , and a discrete Gaussian error  $e \in R_q$  satisfy  $b = a \cdot s + e \pmod{q}$ , and recovering  $s$  is computationally hard. Related lattice problems include the Shortest Vector Problem (SVP):  $\min_{\mathbf{v} \in \mathcal{L}, \mathbf{v} \neq 0} \|\mathbf{v}\|$ , and the RLWE problem: given  $(a, b)$ , finding  $s$  in  $R_q$  is difficult, where  $N$  denotes the polynomial degree,  $q$  the modulus,  $s$  and  $e$  are the secret and error polynomials, and  $\|\cdot\|$  is the Euclidean norm.

### C. RLWE Assumption

The RLWE assumption is a variant of LWE defined over polynomial [40]. Let  $R_q = \mathbb{Z}_q[x]/(x^n + 1)$ , and consider a

secret polynomial  $s(x) \in R_q$ . Given multiple samples of the form

$$t(x) = a(x) \cdot s(x) + e(x) \pmod{q}, \quad (1)$$

where  $a(x)$  is uniformly random and  $e(x)$  is drawn from a discrete Gaussian, it is computationally hard to recover  $s(x)$  without knowledge of  $e(x)$ . This hardness underlies the security of the proposed threshold signature scheme.

### D. Discrete Gaussian Sampling and Rejection Sampling

We elaborate on two typical sampling schemes, which are the discrete Gaussian sampling and the rejection sampling.

**Discrete Gaussian Sampling.** Discrete Gaussian sampling is essential for key generation and signature computation. Given mean  $\mu$  and standard deviation  $\sigma$ , a discrete Gaussian distribution over integers is defined as

$$D_{\mathbb{Z}, \sigma, \mu}(x) = \frac{\exp(-\pi(x - \mu)^2/\sigma^2)}{\sum_{y \in \mathbb{Z}} \exp(-\pi(y - \mu)^2/\sigma^2)}, \quad x \in \mathbb{Z}. \quad (2)$$

This distribution ensures that private keys and noise polynomials have appropriate randomness, which is crucial for resisting lattice attacks and ensuring security in the RLWE setting [40].

**Rejection Sampling.** To generate samples conforming to the discrete Gaussian distribution, rejection sampling is a typical scheme. Each participant repeatedly draws a candidate  $x$  from a proposal distribution (e.g., uniform or normal), then accepts it with probability

$$p = \exp\left(-\frac{(x - \text{round}(x))^2}{2\sigma^2}\right), \quad (3)$$

until the desired number of samples is obtained. This method guarantees that the resulting discrete samples accurately approximate the Gaussian distribution, which is critical for key generation, signing, and shared refresh operations in the QR-LBTS scheme.

## IV. QR-LBTS SCHEME

In this section, we mainly introduce the proposed scheme, including the following parts: system model and design goals, security model, QR-LBTS system structure, QR-LBTS system module and the security analysis.

### A. System Model and Design Goals

The QR-LBTS framework operates in a multi-party computation environment, aiming to provide secure and efficient threshold signature generation and verification. The system involves six core entities, as illustrated in Fig. 2:

- **Message:** The original information to be transmitted among participants.
- **Vehicle:** Entities that receive messages, perform sampling, and prepare partial shares.
- **Sampling:** Mechanism to select required data elements from the message for share generation.
- **Shares:** Secret fragments produced by participants for later aggregation and verification.
- **Ciphertext:** Encrypted form of shares or messages to ensure confidentiality during transmission.

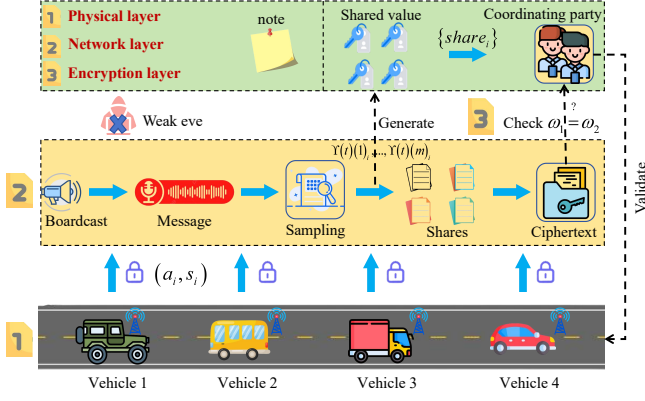


Fig. 2. Our proposed lattice-based framework for vehicular networks.

- **Coordinating Party:** Aggregates shares, validates commitment values, and performs conformance checks to produce the global signature.

The design of QR-LBTS focuses on achieving both efficiency and security:

- **Efficiency:** The framework supports low-latency threshold signature generation and verification in large-scale distributed systems. This is realized by reducing communication rounds through local weighted computations and incremental merging, optimizing polynomial multiplications with NTT, and accelerating critical operations such as discrete Gaussian rejection sampling and shared refresh. As a result, computational and communication overheads remain minimal even with a large number of participants.
- **Security:** QR-LBTS is constructed to withstand both classical and post-quantum adversaries, including unauthorized data injection, man-in-the-middle attacks, temporary secret leakage attack and side-channel attack. By employing RLWE-based lattice constructions, Crystals-Dilithium signature primitives, discrete Gaussian sampling, and a robust shared refresh mechanism, the protocol guarantees unpredictability of private keys, confidentiality of partial signatures, and overall authenticity, integrity, and fault tolerance in multi-party computations.

### B. Security Model

We define the security of the QR-LBTS scheme by considering a Quantum Polynomial-Time (QPT) adversary  $\mathcal{A}$  attempting to compromise the confidentiality, integrity, and correctness of the system. The adversary has access to all public parameters and may perform polynomial-time operations modulo  $q$ . The security is captured via the following adaptive game:

**Setup:** The challenger  $\mathcal{C}$  runs the system setup to generate public parameters  $pp$ , including system-wide constants, modulus  $q$ , Gaussian parameters, and threshold configuration. Participants generate their private keys and associated public keys using discrete Gaussian sampling and compute masked MAC message shares. Partial signature components are prepared but not revealed.

TABLE III  
SYMBOL DESCRIPTION

| Symbol          | Description                                                                           |
|-----------------|---------------------------------------------------------------------------------------|
| $\lambda$       | Security parameter controlling the overall security level.                            |
| $n$             | Polynomial dimension in the lattice-based ring.                                       |
| $q$             | Prime modulus for polynomial arithmetic over $\mathbb{Z}_q[x]/(x^n + 1)$ .            |
| $s_i, \alpha_i$ | Participant $i$ 's private and random polynomials used in KeyGen and signing.         |
| $t_i$           | Public key of participant $i$ derived from $s_i$ and $\alpha_i$ .                     |
| $\sigma(\xi)_i$ | MAC message share of participant $i$ at index $\xi$ in the MAC Message Sharing phase. |
| $\lambda_i$     | Lagrange interpolation coefficient for aggregating partial shares/signatures.         |
| $\mu, \sigma$   | Mean and standard deviation for Gaussian sampling.                                    |
| $y_i$           | Random polynomial sampled during signing.                                             |
| $c_i$           | Challenge bit in signature generation and verification.                               |
| $z_i$           | Partial signature of participant $i$ .                                                |
| $\eta$          | Noise parameter in RLWE sampling.                                                     |
| $\sigma(\xi)$   | Aggregated global MAC value at index $\xi$ .                                          |
| $t_s$           | Signing threshold.                                                                    |
| $t_v$           | Verification threshold.                                                               |

**Queries:**  $\mathcal{A}$  may adaptively query the following oracles:

- **PartialShare:**  $\mathcal{A}$  may obtain masked MAC message shares of a subset of participants,
- **PublicKey:**  $\mathcal{A}$  may obtain public keys derived from private keys via discrete Gaussian sampling,
- **PartialSign:**  $\mathcal{A}$  may obtain partial signatures for chosen messages,
- **Refresh:**  $\mathcal{A}$  may observe refreshed shares after randomization rounds.

**Challenge:**  $\mathcal{A}$  attempts to either:

- recover the original message or secret shares from fewer than the required threshold of participants,
- infer any participant's private key from public information,
- or forge a valid global signature that passes verification without collecting enough legitimate partial signatures.

**Guess:**  $\mathcal{A}$  outputs a candidate value representing either a reconstructed share, private key, or forged signature. The adversary wins if the guess allows recovery of secrets or passes verification, which would indicate a breach of QR-LBTS security.

**Security Definition:** QR-LBTS is secure if, for all QPT adversaries  $\mathcal{A}$ , the probability of winning the above game is negligible in the security parameter  $\lambda$ , i.e.,

$$\Pr[\text{adversary wins}] \leq \text{negl}(\lambda). \quad (4)$$

This unified model captures all algorithmic phases of QR-LBTS: masked MAC Message Sharing, discrete Gaussian KeyGen, Share Refresh, Rejection Sampling, partial Sign, and global verification. The security relies on the hardness of the RLWE problem, the statistical indistinguishability of discrete Gaussian sampling, and threshold enforcement in combining partial shares and signatures.

### C. QR-LBTS System Structure

The QR-LBTS framework is organized as a collaborative multi-party threshold-signature workflow, consisting of *Setup*, *Local Neighbor Weighted Computation (NWC)*, *KeyGen*, *Shared Refresh*, *MAC Message Sharing*, *Rejection Sampling*, *Sign*, and *Verify*. The main symbols used in the QR-LBTS workflow are summarized in Table III.

*Setup* initializes the system parameters and participant states, while *Local NWC* performs neighborhood-based pre-aggregation to reduce communication overhead before cryptographic processing. *KeyGen* establishes the local key material for each participant, and *Shared Refresh* updates intermediate shares to enhance robustness against long-term leakage. *MAC Message Sharing* generates authenticated intermediate values for consistency checking, whereas *Rejection Sampling* produces statistically suitable discrete samples for secure lattice-based operations. Based on these components, *Sign* combines at least  $t_s$  valid partial signatures into a global signature, and *Verify* supports adaptive validation through the verification threshold  $t_v$ , with fallback to full-threshold checking when needed.

Overall, this staged design provides an end-to-end workflow for secure, scalable, and post-quantum-resistant threshold authentication in dynamic VANET environments. Detailed descriptions of each stage are presented in the following subsection.

### D. QR-LBTS System Module

1) **Setup**: The system is initialized with security parameter  $\lambda$ , modulus  $q$ , and lattice dimension  $n$ . A cyclic group structure and the associated polynomial ring  $\mathbb{Z}_q[x]/(x^n + 1)$  are fixed for subsequent operations. Each participant  $P_i$  is assigned a unique identifier  $i \in \{1, \dots, n\}$  and initializes local randomness seeds for sampling. During the setup stage, private key shares  $s_i$ , random polynomials  $\alpha_i \sim \text{Uniform}(0, q)$ , and error terms  $e_i$  are prepared for use in KeyGen. For message authentication, intermediate values  $\sigma(\xi)_i$  and the global MAC value  $\sigma(\xi)$  are defined, where the hash output  $c_i = H(\{\sigma(\xi)_i\})$  serves as the challenge in later signing. The random samples  $y_i$  drawn from rejection sampling are reserved for computing partial signatures  $z_i = y_i + s_i \cdot c_i$ , which are then aggregated into the final signature  $z$ . Verification relies on the public parameters  $\alpha_i$ ,  $t_i = (\alpha_i \cdot s_i + e_i) \bmod q$ , and the relation  $\omega_1 = \text{ntt}(\alpha_i \cdot z)$  versus  $\omega_2 = \text{ntt}(t_i \cdot c_i)$ .

2) **Local NWC Scheme**: To reduce computational overhead, we introduce a local NWC scheme as an optimization layer between the *Setup* and *KeyGen* phases. Here, the neighborhood relation in NWC is interpreted as a temporary logical adjacency within a short local aggregation window, rather than as a persistent physical road-topology assumption. In practice, the temporary local participant set can be formed from the currently reachable and authentication-eligible nodes within the current communication window.

More generally, let  $\mathcal{N}_i(t)$  denote the temporary neighbor set (or selected local participant set) of participant  $P_i$  at

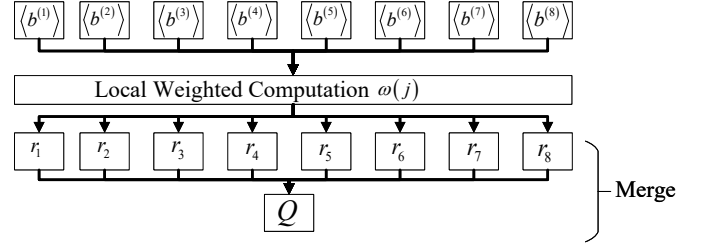


Fig. 3. Illustrative local NWC overlay for temporary participant aggregation.

aggregation epoch  $t$ . Then the local weighted aggregation can be written as

$$r_i(t) = \sum_{j \in \mathcal{N}_i(t) \cup \{i\}} \omega_{ij}(t) \langle b^{(j)}(t) \rangle, \quad (5)$$

where  $\omega_{ij}(t)$  is the time-varying aggregation weight associated with participant  $j$  during epoch  $t$ . This expression captures the dynamic availability of neighboring participants caused by mobility and topology variation in VANETs.

For clarity of exposition and controlled comparison, we further consider a simplified ordered local-participant case in which participant  $i$  interacts with two adjacent logical neighbors, denoted by  $\langle b^{(i-1)} \rangle$  and  $\langle b^{(i+1)} \rangle$ . Under this special case, the local weighted computation becomes

$$r_i = \omega_{i-1} \cdot \langle b^{(i-1)} \rangle + \omega_i \cdot \langle b^{(i)} \rangle + \omega_{i+1} \cdot \langle b^{(i+1)} \rangle,$$

where the  $i-1/i+1$  relation represents logical adjacency in the temporary ordered set rather than a fixed physical neighbor relation in the road network. Note that the illustrative setting  $K = 8$  in Fig. 3 follows [17] for controlled comparison and visualization, and should not be interpreted as implying a fixed VANET topology or a globally stable neighborhood size in practical deployment. Under this simplified ordered local-participant case, the computation process for participants 1 to 8 is given in Eq. (6):

$$\begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_7 \\ r_8 \end{bmatrix} = \begin{bmatrix} \omega_1 & \omega_2 & \cdots & 0 & \cdots & 0 \\ \omega_1 & \omega_2 & \omega_3 & \cdots & \ddots & \vdots \\ 0 & \omega_2 & \omega_3 & \omega_4 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & 0 & \omega_6 & \omega_7 & \omega_8 \\ 0 & \cdots & \cdots & 0 & \omega_7 & \omega_8 \end{bmatrix} \begin{bmatrix} \langle b^{(1)} \rangle \\ \langle b^{(2)} \rangle \\ \langle b^{(3)} \rangle \\ \langle b^{(4)} \rangle \\ \vdots \\ \langle b^{(8)} \rangle \end{bmatrix}. \quad (6)$$

Algorithm 1 provides the corresponding illustrative procedure.

Importantly, these merging rounds are performed within a short local aggregation epoch associated with the current authentication process, rather than as a long-lived consensus procedure over a permanently stable vehicular topology. Therefore, the scheme only requires temporary availability of a local participant set during the current round, which is more compatible with the transient nature of VANET interactions. Here,  $\parallel$  denotes a local merge operation between intermediate aggregation results.

In each round of computation, participants merge their local results progressively until the global computation result is

**Algorithm 1** Local NWC Scheme

---

```

1: Input: Local data  $\langle b^{(j)} \rangle$  and aggregation weights
2: Output: Global result  $Q$ 
3: All participants in the temporary local set prepare local
   data  $\langle b^{(j)} \rangle$  and aggregation weights
4: Generate:
5: for each participant  $P_i$  do
6:   Compute local weighted result:
       
$$r_i = \omega_{i-1} \cdot \langle b^{(i-1)} \rangle + \omega_i \cdot \langle b^{(i)} \rangle + \omega_{i+1} \cdot \langle b^{(i+1)} \rangle$$

7: end for
8: Perform step-by-step merging
9:   First round: Merge  $r_i$  results pairwise
10:  Second round: Merge intermediate results
11:  Third round: Final merge to obtain  $Q$ 
12: Output  $Q$  and broadcast it within the current local aggregation set

```

---

obtained. Specifically, the first round merges adjacent intermediate results, the second round further combines the first-round outputs, and the third round produces the global result  $Q$ . To illustrate the procedure, we give the following example:

- 1) First round:  $r'_1 = r_1 \| r_2$ ,  $r'_2 = r_3 \| r_4$ ,  $r'_3 = r_5 \| r_6$ ,  $r'_4 = r_7 \| r_8$ .
- 2) Second round:  $r''_1 = r'_1 \| r'_2$ ,  $r''_2 = r'_3 \| r'_4$ .
- 3) Third round:  $Q = r''_1 \| r''_2$ .

Compared to the traditional method, our computation-merging strategy reduces the communication rounds to 10. Under the same illustrative setting, this is comparable to the model in [17], while still serving as an effective local pre-aggregation mechanism for the subsequent cryptographic stages. In summary, the NWC scheme acts as a pre-aggregation step that balances local computation and global synchronization, enabling subsequent cryptographic phases—such as KeyGen, Shared Refresh, and MAC Message Sharing—to operate on efficiently fused and noise-tolerant data, thereby improving the overall stability and scalability of the distributed protocol.

3) **KeyGen:** In this phase, each participant  $P_i$  generates a private key  $S_i$ , the security of which relies on using the RLWE sampling method to sample from a discrete Gaussian distribution. Specifically, the private key generation process is as follows:  $s_i = \text{RLWE}(0, \eta, n)$ , where  $\eta$  is the coefficient range that controls the standard deviation of the sampling, and  $n$  is the polynomial dimension. Then, each participant  $P_i$  generates a random polynomial  $\alpha_i$ , whose coefficients are sampled from a uniform distribution modulo  $q$ :  $\alpha_i \sim \text{Uniform}(0, q)$ , where  $q$  is a large prime, used as the modulus to ensure the correctness of polynomial operations and the structural integrity of the finite field. Participant  $P_i$  computes the corresponding public key  $t_i = (\alpha_i \cdot s_i + \text{RLWE}(0, \eta, n)) \bmod q$ . Here,  $\cdot$  denotes the polynomial multiplication operation. This computation produces the participant's encryption part. The term  $\text{RLWE}(0, \eta, n)$  is used to generate the error component, introducing moderate noise to ensure security and resistance to attacks. Finally, the key pair  $(a_i, s_i)$  is obtained and shared

**Algorithm 2** Shared Refresh Protocol

---

```

1: Input: Shares  $\{sh_i\}$ 
2: Output: Refreshed shares  $\{rs_i\}$ 
3: Initialize: shares =  $\{sh_i\}$ 
4: Generate random values: random_share $i$ 
5: Modify: Initialize  $ms_i = \{\}$ 
6: for each  $i$ ,  $sh_i$  in shares do
7:    $rs_i = \text{rv}[i]$ 
8:    $ms_i = (sh_i + rs_i) \bmod q$ 
9:    $rm_i \sim \text{Uniform}(0, q)$ 
10:   $m's_i = (ms_i \times rm_i) \bmod q$ 
11:   $ms[i] = ms_i$ 
12: end for
13: Multiple rounds of calculations: Initialize  $rs_i = \{\}$ 
14: for each  $i$ ,  $ms_i$  in modified_shares do
15:   for  $j = 1$  to 5 do
16:     $ra_i \sim \text{Uniform}(0, q)$ 
17:     $m''s_i = (ms_i + ra_i) \bmod q$ 
18:     $rm\_new_i \sim \text{Uniform}(0, q)$ 
19:     $m'''s_i = (m's_i \times rm\_new_i) \bmod q$ 
20:   end for
21:    $rs[i] = m'''s_i$ 
22: end for
23: return  $rs_i, r's_i$ 

```

---

with other participants.

4) **Shared Refresh:** In this phase, the calculation complexity is increased by performing complex transformations and modifications on each shared value, thereby enhancing attack resistance. The details are as described in Algorithm 2. In specific, suppose a set of shared values  $\{sh_i\}$  is obtained from the previous phase, with each  $sh_i$  containing  $n$  elements, representing the shared value of the  $i$ -th participant, originating from the original secret sharing. For each shared  $rs_i$ , a random value  $rs_i$  is generated, sampled from a uniform distribution modulo  $q$ :  $rs_i \sim \text{Uniform}(0, q)$ . Then, the shares are modified by performing a series of complex modular operations. First, an addition operation is performed:  $ms_i = (sh_i + rs_i) \bmod q$ , followed by a multiplication operation to increase the computational complexity:  $m's_i = (ms_i \times rm_i) \bmod q$ , where  $rm_i$  is a value randomly sampled modulo  $q$ . Subsequently, multiple rounds of computation are performed: for each modified share, multiple rounds of modular addition and modular multiplication are executed. Each round of operations is given by

$$rs_i = (ms_i + ra_i) \bmod q, \quad (7)$$

followed by

$$rs'_i = (rs_i \times rm\_new_i) \bmod q. \quad (8)$$

These steps are repeated multiple times (executed 5 times in the code), with new random addends and multiplication factors used in each iteration to increase the complexity of the operations. Finally, each shared value  $sh_i$  is transformed into a  $rs_i$ , thereby increasing computational complexity and enhancing security.

5) **MAC Message Sharing:** To generate shared MAC values for each participant and perform the necessary computations to ensure message integrity, each participant  $P_i$  obtains the message, and extracts shares from its shared MAC value, denoted as  $\Upsilon(t)(1)_i, \dots, \Upsilon(t)(m)_i$ , and the pre-processed MAC key share, denoted as  $\alpha_i = (\alpha(1)_i, \dots, \alpha(m)_i)$ . As shown in Fig. 1, the participants jointly generate a random vector, denoted as  $r(1), \dots, r(v)$ , and compute  $tr = \sum_{j=1}^v r(j) \cdot t(j) \bmod q$ , where,  $t(j)$  is the message portion of participant  $j$ , and  $r(j)$  is the randomly sampled coefficient. For each  $\xi \in [m]$ , each participant  $P_i$  performs the following calculation:

- 1) We split the MAC share  $\Upsilon(t)(\xi)_i$  into  $(\Upsilon(t)(\xi)_i, \dots, \Upsilon(v)(\xi)_i)$ , and compute  $\Upsilon(\xi)_i = \sum_{j=1}^v r(j) \cdot \Upsilon(t(j))(\xi)_i \bmod q$ .
- 2) Then, we calculate  $\sigma(\xi)_i = \lambda_{i,[n]} \cdot (\Upsilon(\xi)_i - \alpha(\xi)_i \cdot tr) \bmod q$ , where  $\lambda_{i,[n]} = \prod_{j \in [n], j \neq i} \frac{j}{j-i}$  is the Lagrange coefficient.

Each participant  $P_i$  computes  $c_i = H(\{\sigma(\xi)_i\}_{\xi=1}^m)$ , where  $H$  is the hash function, and the generated hash value  $c_i$  is used as the current participant's MAC value. Each participant broadcasts  $\{\sigma(\xi)_i\}_{\xi=1}^m$  and verifies the following consistency condition:  $H(\{\sigma(\xi)_{\bar{i}}\}_{\xi=1}^m) = c_{\bar{i}}$ , for all  $\bar{i} \neq i$ . If the consistency check fails, output 0 and terminate. Finally, all participants compute the globally shared signature value  $\sigma(\xi) = \sum_{i=1}^n \sigma(\xi)_i \bmod q$ , for  $\xi \in [m]$ . The final MAC value share  $\sigma(\xi)$  is used for subsequent verification and usage, i.e., if all  $\sigma(\xi) = 0$  for each  $\xi \in [m]$ , output 1, indicating that all MAC values are valid. Otherwise, output 0, indicating the discovery of inconsistency or fraud.

6) **Reject Sampling:** In this phase, we raise computational complexity by applying statistically-aware sampling and validation steps to each secret-dependent value, thereby improving resistance to statistical and side-channel attacks. To sample from a discrete Gaussian distribution, generate random numbers tied to the private key and use rejection sampling to ensure the produced samples conform to the target distribution. We carefully show the process in Algorithm 3. The parameters received by the participants during the sampling phase are given, such as mean  $\mu$ , standard deviation  $\sigma$ , and sample size  $n$  (corresponding to the polynomial dimension). These parameters are provided by the protocol and determine the distribution characteristics of the samples. To sample from the discrete Gaussian distribution and generate random numbers related to the private key, this process is based on the standard normal distribution and uses the rejection sampling method to convert the sample into discrete values, denoted as  $sa_i \sim \mathcal{N}(\mu, \sigma)$  for each  $P_i$ . Each participant  $P_i$ , given the mean  $\mu$  and standard deviation  $\sigma$ , independently generates a sample set of size  $n$  from the normal distribution, denoted as  $sa_i = \text{Sample from normal distribution}(\mu, \sigma, n)$ . Then, each sample is rounded to map it to the integer space, ensuring it conforms to the discrete nature of the key. This step produces the discretized sample, denoted as  $ds_i = \text{round}(sa_i)$ . For each sample, participant  $P_i$  calculates the acceptance probability  $p_i$  based on the difference between the sample value and the

---

**Algorithm 3** Rejection Sampling Protocol

---

- 1: **Input:** Mean  $\mu$ , standard deviation  $\sigma$ , sample size  $n$
  - 2: **Output:** Discrete  $as$
  - 3: **Initialize:** Set  $\mu$ ,  $\sigma$ , and  $n$
  - 4: **Sampling Phase:**
  - 5: **for** each participant  $P_i$  **do**
  - 6:   Generate samples:  $sa_i \leftarrow \text{Sample from normal distribution}(\mu, \sigma, n)$
  - 7:   Discretize samples:  $ds_i \leftarrow \text{round}(sa_i)$
  - 8:   Compute acceptance probability:
$$p_i \leftarrow \exp\left(-\frac{(sa_i - ds_i)^2}{2\sigma^2}\right)$$
  - 9:   **repeat**
  - 10:     Generate  $r_i \leftarrow U(0, 1)$
  - 11:     Set  $ac_i \leftarrow (r_i < p_i)$
  - 12:     **if**  $ac_i$  is False **then**
  - 13:       Resample: regenerate  $sa_i$  and recompute
  - 14:     **end if**
  - 15:     **until**  $ac_i$  is True
  - 16:      $as_i \leftarrow ds_i$
  - 17: **end for**
  - 18: **return**  $as_i$  for all  $P_i$
- 

discretized sample value, as shown in Equation (9):

$$p_i = \exp\left(-\frac{(sa_i - ds_i)^2}{2\sigma^2}\right). \quad (9)$$

Each participant  $P_i$  generates a random number from a uniform distribution to decide whether to accept the sample: generate a random number  $r_i$  from the uniform distribution  $U(0, 1)$ , denoted as  $r_i \sim U(0, 1)$ . If the random number  $r_i$  is less than the acceptance probability  $p_i$ , the sample is accepted, otherwise, the sample is rejected and resampled, as determined by  $ac_i = (r_i < p_i)$ . Each participant  $P_i$  repeats the sampling and acceptance steps until the accepted sample meets the required criteria. The rejection sampling process ends only when all  $P_i$  acceptance criteria are met. If all participants pass the acceptance judgment, the rejection sampling process ends, and the accepted discretized samples are used as part of the key, passing to subsequent operations, denoted as  $as_i = ds_i$  if  $ac_i = 1$ .

7) **Sign:** We propose the distributed signature protocol in which participants generate partial signatures of a message and then merge them into the final signature with a threshold-based aggregation. Specifically, each participant  $P_i$  receives parameters from the MAC Message Sharing stage, the Shared Refresh stage, and the KeyGen stage:  $message$ ,  $cv$ , and  $(s_i, a_i)$ , as input to generate the partial signature of the message.

1) **Commitment check:** Each participant  $P_i$  computes the commitment value  $\text{commitment}(message)$  and compares it with the received  $cv$ . If they do not match, the signing process terminates for this participant.

2) **Partial signature generation:** Each participant samples  $y_i \sim \text{RS}(0, 2^d, n)$  from the discrete Gaussian distribution. Then, compute  $\omega_i = \text{NTT}(\alpha_i \times y_i)$  and select challenge bit

---

**Algorithm 4** Verify Protocol
 

---

```

1: Input:  $t_s$  (signing threshold),  $t_v$  (verification threshold),
    $message, z, c_i, \alpha, cv$ 
2: Output: 1 (valid) or 0 (invalid)
3: Each  $P_i$  receives  $message, cv$ , challenge  $c_i$ , signature  $z$ ,
   and  $t_i$ 
4: Step 1: Verify commitment:
5:  $ck \leftarrow (\text{commitment}(message) == cv)$ 
6: if  $ck$  is False then
7:   Abort and output 0
8: end if
9: Step 2: Compute verification values:
10:  $\omega_1 \leftarrow \text{ntt}(\text{poly\_mul\_mod}(\alpha, z))$ 
11:  $\omega_2 \leftarrow \text{ntt}(\text{poly\_mul\_mod}(t_i, c_i))$ 
12:  $\text{verify\_check}_i \leftarrow (\text{mod}_q(\omega_1 - \omega_2) == 0)$ 
13: Step 3: Adaptive threshold validation:
14: Collect  $\text{verify\_check}_i$  results from all participating veri-
   fiers
15: if number of True  $\text{verify\_check}_i \geq t_v$  then
16:   Return: 1 (signature is valid)
17: else
18:   Optionally trigger full verification with at least  $t_s$ 
   participants
19:   if full verification passes then
20:     Return: 1
21:   else
22:     Return: 0 (signature is invalid)
23:   end if
24: end if

```

---

$c_i \sim \text{Uniform}([-1, 0, 1], n, p)$ . Finally, generate the partial signature  $z_i = y_i + (s_i \times c_i)$ .

3) Broadcast partial signatures: Each participant broadcasts  $(z_i, c_i)$  to other participants.

4) Threshold aggregation: Once at least  $t_s$  participants have shared their partial signatures, combine them using Lagrange interpolation:  $z = \sum_{i \in S} \lambda_i \times z_i \text{ mod } q$ , where  $S$  is the subset of signers with  $|S| \geq t_s$  and  $\lambda_i$  are the Lagrange coefficients. This ensures that the final signature is valid even if not all participants contribute, reflecting the signing threshold  $t_s$ . The final signature  $z$  is thus jointly produced by the participating signers and ready for verification.

8) **Verify:** Each participant  $P_i$  receives  $message, cv, c_i, \alpha, z$ , and  $t_i$  from the relevant protocol stages. Verification proceeds as follows:

1) Commitment check: Compute  $\text{commitment}(message)$  and compare with  $cv$ . If the mismatch occurs, terminate verification for this participant.

2) Compute verification values:

$$\begin{aligned}
 \text{verify\_check}_i &= \text{mod}_q(\omega_1 - \omega_2) \\
 &= \text{mod}_q(\text{ntt}(\text{poly\_mul\_mod}(\alpha, z) \\
 &\quad - \text{poly\_mul\_mod}(t_i, c_i))). \quad (10)
 \end{aligned}$$

3) Adaptive threshold validation: The signature is considered valid if at least  $t_v$  participants ( $t_v \leq t_s$ ) report  $\omega_1 = \omega_2$ . If fewer than  $t_v$  participants pass, trigger full verification requiring at least  $t_s$  verification participants.

4) Output result: If the threshold condition is met, output "valid". Otherwise, output "invalid" and optionally invoke complete verification. To maintain consistency and security between signing and verification, the thresholds must satisfy  $1 \leq t_v \leq t_s \leq n$ . In practice,  $t_v$  can be defined as a fraction of  $t_s$ , such that  $t_v = \lfloor \beta \cdot t_s \rfloor$  with  $0 < \beta \leq 1$ , where  $\beta = 1$  enforces strict full-threshold verification and smaller  $\beta$  values enable early validation with reduced computational cost. Regardless of  $\beta$ , the correctness of the final signature remains bound by the  $t_s$ -out-of- $n$  security guarantee.

This design integrates the adaptive threshold mechanism directly into both signing and verification, allowing flexible participation while preserving the  $t_s$ -out-of- $n$  security guarantees.

### E. Security Analysis

We analyze the security of our QR-LBTS scheme under the unified security model. Let  $\mathcal{A}$  be a quantum polynomial-time adversary with access to public parameters and polynomial-time oracles. The security relies on the hardness of lattice problems—RLWE and the statistical properties of discrete Gaussian sampling.

*Theorem 1 (Confidentiality of Partial Shares and Keys).* Under the RLWE assumption and statistical indistinguishability of discrete Gaussian samples, any adversary  $\mathcal{A}$  with fewer than the threshold number of shares cannot recover private keys or MAC message shares, except with negligible probability.

*Proof.* Each share is masked with independent random vectors and combined using Lagrange coefficients. Private keys ( $s_i$ ) are sampled from a discrete Gaussian and public keys are computed as  $t_i = \alpha_i \cdot s_i + e_i \text{ mod } q$ . By the RLWE hardness, recovering  $s_i$  from  $t_i$  is infeasible. The randomness of masks and share combinations ensures that any subset of fewer than  $n$  shares provides no meaningful information. Therefore, the probability of reconstructing secrets is negligible:  $\Pr[\text{recover } s_i \text{ or MAC share}] \leq \text{negl}(\lambda)$ .  $\square$

*Theorem 2 (Robustness of Share Refresh and Rejection Sampling).* After multiple rounds of share refresh and discrete Gaussian rejection sampling, refreshed shares and sampled values remain statistically indistinguishable from uniform or ideal Gaussian distributions, preventing biasing or leakage by an adversary.

*Proof.* Refreshed shares are computed via modular addition and multiplication with independent random vectors. Rejection sampling ensures that accepted discrete samples closely approximate Gaussian distributions. Any adversary attempting to bias or distinguish the samples would need to break the statistical properties of discrete Gaussian distributions or invert the modular transformations, which is negligible under the RLWE assumption:  $\Pr[\text{bias or predict samples}] \leq \text{negl}(\lambda)$ .  $\square$

*Theorem 3 (Unforgeability of Partial and Global Signatures).* Without access to at least  $t_s$  valid partial signatures, a quantum adversary cannot forge a valid global signature.

*Proof.* Partial signatures  $z_i$  are generated using private keys, random masks, and challenge-dependent coefficients. The threshold property guarantees that fewer than  $n$  legitimate partial signatures cannot reconstruct a correct global signature. Forging  $z$  requires either predicting

discrete Gaussian samples or solving RLWE instances to recover  $s_i$ . Both of which are negligible in probability:  $\Pr[\text{forge global signature without threshold}] \leq \text{negl}(\lambda)$ .  $\square$

*Theorem 4 (Verification Integrity).* *The usage of public keys, challenge bits, and the global signature in verification ensures that no adversary can pass the modular checks without sufficient legitimate partial signatures.*

*Proof.* Verification computes modular equalities derived from public keys and combined partial signatures. Any incorrect or incomplete signature fails these checks unless the adversary can invert RLWE transformations or guess discrete Gaussian samples correctly, which occurs with negligible probability:  $\Pr[\text{pass verification with invalid signature}] \leq \text{negl}(\lambda)$ .  $\square$

*Parameter Guidance.* The selection of modulus  $q$ , lattice dimensions  $n, m$ , Gaussian width  $\eta$ , and threshold  $n$  such that  $n \cdot \eta \gg \sqrt{\log n}$  and  $q \gg \max(\text{coefficients})$  ensures RLWE hardness and statistical indistinguishability.

Under the above assumptions, our QR-LBTS scheme achieves quantum-resistant security for shared confidentiality, sample robustness, partial and global signature integrity, and threshold verification.

## V. EXPERIMENT

In this section, we conduct an in-depth analysis of the performance of the key generation, signing, shared refresh, rejection sampling, and other related stages in the proposed scheme.

### A. Overall Performance

The experiment was conducted on a Windows 11 operating system, using a 12th Gen Intel(R) Core(TM) i7-12700H 2.30 GHz processor, with Python 3.10 for simulation. The experiment is conducted in a LAN environment with 16 threads operating in parallel.

Compared with the TSS lattice [41] and Tang et al. [17], our QR-LBTS scheme demonstrates significantly superior computational efficiency. Across different party sizes from 2 to 12, the KeyGen and Sign phases remain within 1.23–3.39 s and 0.18–1.14 s, respectively, highlighting its lightweight and scalable design. The Share Refresh and Reject Sampling also introduce only minimal overhead, further confirming the scheme’s practicality in dynamic multiparty settings. In contrast, the TSS scheme [41] incurs considerably higher computational costs, particularly in the 12-party case, where the KeyGen and Sign phases reach 4.94 s and 12.87 s, respectively. Likewise, Tang et al.’s scheme [17] exhibits substantial overhead, with KeyGen and Sign times of 4.38 s and 17.4 s in the 12-party case.

### B. Performance Analysis

**Parameter Settings.** The parameter settings in Table IV are selected with reference to Tang et al. [17] and are designed to achieve a security strength of 123 classical bits and 112 quantum bits. The selection of these parameters takes into account the security requirements under both classical and quantum

TABLE IV  
PARAMETER SETTING

| Parameters | $N$ | $q$       | $d$ | $\tau$ | $\eta$ | $(K, l)$ |
|------------|-----|-----------|-----|--------|--------|----------|
| Values     | 256 | 8,380,417 | 13  | 60     | 2      | (4, 4)   |

threat models, ensuring that the proposed scheme provides robust protection against both traditional and quantum attacks. The specific meanings and explanations of these parameters can be found in the detailed description in Table III.

Our online-phase experiments were implemented on the Alibaba Cloud platform. In this work, we focus on the configuration and performance evaluation of the proposed authentication algorithm in the existing deployment environment, rather than on the dedicated design or reimplementing of the underlying wireless communication stack itself. Accordingly, the current evaluation should be interpreted as a protocol-level assessment of QR-LBTS at the authentication layer. Under this evaluation setting, our online-phase experiments were conducted in two network environments: LAN and WAN. In the LAN network, we set up 12 nodes in a group for the experiment, while in the WAN network, each group consisted of three nodes located in different cities, simulating inter-city network communication. In order to simulate network latency across different environments, we deployed our tests on the Alibaba Cloud platform. By tuning environment variables, we were able to replicate the characteristics of both LAN and WAN environments.

In the LAN environment, we set the bandwidth to 400 Mbps and the delay to 0.3 ms, reflecting the characteristics of low latency and high bandwidth typical of local area networks. In the WAN environment, the bandwidth was set to 352 Mbps, and the delay was set to 40 ms, simulating the higher latency and slightly lower bandwidth found in wide area networks. These experimental setups aim to accurately assess the algorithm’s performance under different network conditions, particularly focusing on the impact of network bandwidth and latency on communication efficiency and response time.

**Computational overhead.** We compare the proposed QR-LBTS scheme with Tang et al. ([17]), TSS [41], and the ML-DSA baseline in terms of the execution time of KeyGen, Sign, Share Refresh, and Reject Sampling under different numbers of participants, as shown in Figs. 4a–4d. The results show that the proposed QR-LBTS scheme provides the most favorable overall efficiency–scalability trade-off for multiparty threshold signing. In particular, in the Sign phase, QR-LBTS consistently achieves the lowest runtime among all compared schemes, and its growth trend remains much smoother as the number of participants increases, demonstrating clear advantages in signing efficiency and scalability. In the KeyGen phase, the proposed scheme also maintains competitive performance and remains markedly more efficient than the ML-DSA baseline, while avoiding the pronounced growth observed in Tang et al. ([17]) as the system scale increases. For Share Refresh, although ML-DSA exhibits a lighter runtime profile, QR-LBTS shows a nearly stable execution time across different party sizes and remains substantially

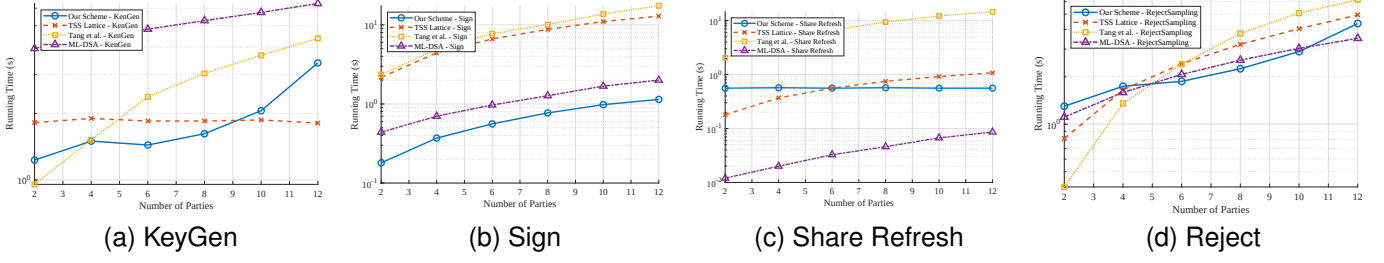


Fig. 4. Computational overhead of different protocols in the KeyGen, Sign, Share Refresh, and Reject phases.

more efficient than Tang et al. ([17]), indicating better consistency in threshold maintenance. A similar observation holds for the Reject Sampling phase: while ML-DSA is slightly faster in absolute runtime, the proposed QR-LBTS scheme still preserves competitive efficiency and outperforms Tang et al. ([17]) at larger party sizes. Compared with TSS [41], QR-LBTS further avoids the sharp overhead escalation in the Sign phase, which is especially important for large-scale and latency-sensitive threshold-signature deployments. Overall, these results suggest that the proposed QR-LBTS scheme is better suited to practical multi-party settings, since it achieves lower online signing cost, competitive key-generation efficiency, and more stable scaling behavior across the core protocol phases. Although a visible increase in runtime appears when the number of participants reaches 10, further analysis shows that this fluctuation is caused by an isolated signature verification failure during the multi-party threshold signing process. This temporary overhead increase does not change the overall trend, but rather reflects the robustness cost incurred to preserve correctness in practical threshold-signature execution.

**Communication cost.** Table V summarizes the communication costs of the compared protocols. Tang et al.’s scheme [17] has a relatively simple structure and is suitable for small-scale scenarios, but its Sign and Reject costs grow with the security parameter and the number of signing-related components. The TSS lattice scheme [41] provides stronger threshold functionality, but its communication expressions involve quadratic terms such as  $n^2$  and  $k^2m$ , which may limit its scalability in large-scale deployments.

The ML-DSA-based scheme introduces a modern Dilithium-style lattice baseline, with KeyGen cost determined by the matrix-vector structure  $(k_d l_d + k_d + l_d)n$  and ShareRefresh cost growing as  $(k_d + l_d)mn$ . In comparison, the proposed QR-LBTS scheme adopts a more balanced communication structure across the major protocol phases. Its KeyGen and Sign costs are both given by  $kn + n \log n$ , while ShareRefresh and Reject remain linear in the dominant system parameters. Although QR-LBTS is not the strict minimum in every individual phase, it avoids the quadratic communication terms that appear in TSS and maintains a more regular threshold-oriented cost profile than the other baselines. Therefore, QR-LBTS offers a favorable balance between communication efficiency, scalability, and deployment practicality in multiparty authentication scenarios.

Table VI reports the communication rounds required by Tang et al. [17], TSS [41], ML-DSA, and the proposed

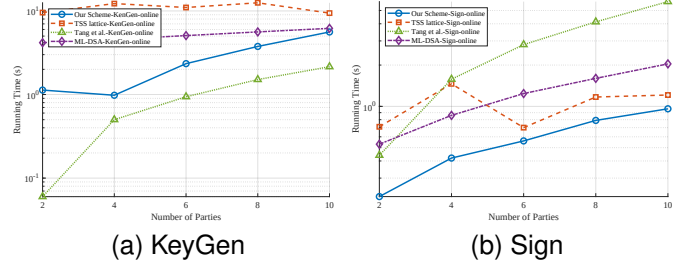


Fig. 5. The running times of the distributed key generation and signing phases over LAN.

scheme to complete the  $(t, n)$  threshold signature. Tang et al.’s scheme requires relatively few rounds in small-scale settings, with KeyGen and Sign depending logarithmically on  $n_q$ , although the signing phase still introduces additional constant interaction rounds. The TSS lattice scheme has higher round complexity, especially in the Sign and Reject phases, where the number of rounds increases with  $n$ ,  $m$ , and  $k$ , which may reduce efficiency in large-scale deployments.

The ML-DSA-based scheme serves as a modern Dilithium-style baseline. Its KeyGen and ShareRefresh phases require  $m$  rounds, and its Sign phase requires  $t + 1$  rounds due to the involvement of  $t$  signers and an additional aggregation/broadcast step. By comparison, the proposed QR-LBTS scheme uses  $k$  rounds for both KeyGen and Sign, while maintaining a fixed 10-round Reject phase. This gives QR-LBTS a more regular communication pattern in the main online stages.

Although the ShareRefresh phase of QR-LBTS requires  $5m$  rounds, this operation is invoked less frequently than online signing in practical deployments. More importantly, QR-LBTS keeps the key-generation and signing procedures structurally simple, while avoiding the heavier round growth of TSS in the online phases. Overall, the proposed scheme provides a better balance between communication efficiency, scalability, and threshold-oriented practicality, making it more suitable for large-scale and dynamic multiparty authentication environments.

**Performance of QR-LBTS in LAN environment.** As illustrated in Fig. 5, we compare the online KeyGen and Sign performance of QR-LBTS with the TSS lattice algorithm [41], Tang et al. [17], and ML-DSA under the LAN setting. Overall, QR-LBTS achieves the most balanced online efficiency among the compared schemes. In the KeyGen phase, QR-LBTS remains within 0.98–5.60 s as the number of parties increases

TABLE V  
COMMUNICATION COST OF EACH PROTOCOL

| Protocol   | KeyGen                                               | ShareRefresh        | Sign                       | Reject                   |
|------------|------------------------------------------------------|---------------------|----------------------------|--------------------------|
| Ref. [17]  | $2\lambda n_q$                                       | $n_q \log_2 n_q$    | $(2(k+1)\lambda + 3k)n_q$  | $2(k+1)\lambda n_q$      |
| TSS [41]   | $k \cdot l + k \cdot m + t \cdot n + n \log q + n^2$ | $n^2 + n \log q$    | $\frac{n^2}{p} + n \log q$ | $k^2 \cdot m + m \log q$ |
| ML-DSA     | $(k_d l_d + k_d + l_d)n$                             | $(k_d + l_d)mn$     | $(k_d + l_d)n + \lambda$   | $n$                      |
| Our Scheme | $k \cdot n + n \log n$                               | $5 \cdot m \cdot n$ | $k \cdot n + n \log n$     | $k \cdot n$              |

TABLE VI  
ROUNDS OF EACH PROTOCOL

| Protocol   | KeyGen       | ShareRefresh   | Sign               | Reject       |
|------------|--------------|----------------|--------------------|--------------|
| Ref. [17]  | $\log_2 n_q$ | $2 \log_2 n_q$ | $2 \log_2 n_q + 5$ | 10           |
| TSS [41]   | $\log n + 1$ | $\log n + 2$   | $n + 2$            | $m + \log k$ |
| ML-DSA     | $m$          | $m$            | $t + 1$            | 10           |
| Our Scheme | $k$          | $5 \cdot m$    | $k$                | 10           |

TABLE VII  
COMPARISON OF DIFFERENT PROTOCOLS ON LAN

| Schemes                 | Runtime (s) | Threshold  |
|-------------------------|-------------|------------|
| Tang et al. Sign [17]   | 0.5         | $t \leq n$ |
| TSS lattice KeyGen [41] | 9.56        | $t = n$    |
| TSS lattice Sign [41]   | 0.71        | $t = n$    |
| ML-DSA Sign             | 0.57        | $t \leq n$ |
| ML-DSA KeyGen           | 4.17        | $t \leq n$ |
| Our Sign                | 0.22        | $t \leq n$ |
| Our KeyGen              | 1.3         | $t \leq n$ |

from 2 to 10, which is substantially lower than TSS lattice at 9.42–12.51 s and also lower than ML-DSA at 4.17–6.19 s. At the 10-party scale, QR-LBTS reduces the KeyGen overhead by approximately  $1.68\times$  compared with TSS lattice and  $1.11\times$  compared with ML-DSA. Although Tang et al. [17] has a lower KeyGen time in small-scale settings, its advantage is offset by much higher signing overhead.

More importantly, QR-LBTS shows a clear advantage in the Sign phase, which is the more frequently executed online operation in practical authentication scenarios. Its signing time increases only from 0.22 s to 0.96 s, remaining below 1 s even with 10 parties. By contrast, TSS lattice ranges from 0.70 s to 1.46 s, Tang et al. [17] increases sharply from 0.44 s to 5.80 s, and ML-DSA grows from 0.53 s to 2.04 s. At 10 parties, QR-LBTS is approximately  $1.26\times$  faster than TSS lattice,  $6.04\times$  faster than Tang et al. [17], and  $2.13\times$  faster than ML-DSA in the Sign phase.

These results indicate that QR-LBTS does not merely optimize a single phase, but provides a more practical trade-off between key generation and signing efficiency. Compared with TSS lattice and ML-DSA, QR-LBTS significantly reduces KeyGen overhead, while compared with Tang et al. [17], it avoids the rapid growth of signing latency as the number of parties increases. Therefore, QR-LBTS offers stronger overall online performance, better scalability, and higher suitability for LAN-based multiparty authentication scenarios.

Table VII compares the performance of different algorithms during the KeyGen and Sign phases under the LAN setting. Since [17] has already compared the distributed rejection sampling process using Garbled Circuits (GC), this process is not repeated in the present study. As shown in the table, our algorithm achieves superior efficiency in the Sign phase, requiring only 0.22 s, which is about 2.3 times faster than Tang et al. [17] and about 2.6 times faster than ML-DSA. In the KeyGen phase, our scheme requires 1.3 s, which is significantly lower than TSS lattice [41] at 9.56 s and ML-DSA at 4.17 s, corresponding to approximately  $7.4\times$  and  $3.2\times$  improvements, respectively. These results indicate that our scheme not only supports flexible threshold access structures with  $t \leq n$ , but also achieves lower online overhead than existing schemes, including Tang et al. [17], TSS lattice [41], and ML-DSA, demonstrating better scalability and practicality in LAN-based multiparty authentication scenarios.

**Performance of QR-LBTS in WAN environment.** In Table VIII, we present the execution times of the protocols under two different threshold settings in the WAN environment. Compared with ML-DSA, QR-LBTS shows clear advantages in all phases. For the (2, 2) setting, our scheme reduces the KeyGen time from 4.81 s to 0.33 s, the Sign time from 1.38 s to 0.12 s, and the ShareRefresh time from 1.44 s to 0.09 s. For the (2, 3) setting, QR-LBTS also remains significantly faster, requiring only 0.55 s for KeyGen, 0.17 s for Sign, and 0.13 s for ShareRefresh, compared with 7.16 s, 1.73 s, and 2.02 s for ML-DSA, respectively. These results show that QR-LBTS maintains much lower latency and better stability under WAN conditions, especially in the online Sign and ShareRefresh phases. This advantage is mainly attributed to the design of our protocol, which shifts most computational overhead to local processing and reduces costly online interactions, thereby improving scalability and practicality in distributed threshold authentication scenarios.

Table IX demonstrates the improvement of our distributed Sign protocol under both LAN and WAN settings compared with GMPC and existing works. Compared with Tang et al. [17], TSS [41], and ML-DSA, our scheme maintains the minimum communication rounds while achieving the best runtime performance. Specifically, our Sign protocol requires only 0.18 s in LAN and 0.17 s in WAN, whereas ML-DSA requires 0.57 s and 1.73 s, respectively. This means that our scheme is about  $3.2\times$  faster than ML-DSA in LAN and about  $10.2\times$  faster in WAN, showing stronger robustness against network-latency variation. These results demonstrate that our scheme provides lower signing latency, fewer communication rounds, and better practicality for distributed threshold signing.

TABLE VIII  
THE RUN TIMES OF OUR PROTOCOL OVER WAN

| Schemes          | Threshold ( $t, n$ ) | KeyGen (s) | Sign (s) | Share Refresh (s) |
|------------------|----------------------|------------|----------|-------------------|
| Tang et al. [17] | (2,2)                | 0.36       | 1.5      | 0.27              |
|                  | (2,3)                | 0.59       | 0.53     | 1.65              |
| TSS lattice [41] | (2,2)                | 8.32       | 3.32     | 6.53              |
|                  | (2,3)                | 4.79       | 2.07     | 3.42              |
| ML-DSA           | (2,2)                | 4.81       | 1.38     | 1.44              |
|                  | (2,3)                | 7.16       | 1.73     | 2.02              |
| Our scheme       | (2,2)                | 0.33       | 0.12     | 0.09              |
|                  | (2,3)                | 0.55       | 0.17     | 0.13              |

TABLE IX  
PERFORMANCE COMPARISON OF OUR SIGN WITH UNIVERSAL MPC, AND  
EXISTING WORKS AT THE 2/3 THRESHOLD SETTING

| Protocol              | Runtime (s) |      | Rounds |
|-----------------------|-------------|------|--------|
|                       | LAN         | WAN  |        |
| GMPC-Sign             | 12.4        | 15.7 | 60     |
| Tang et al. Sign [17] | 0.5         | 1.65 | 15     |
| TSS Sign [41]         | 0.71        | 3.32 | 10     |
| ML-DSA Sign           | 0.57        | 1.73 | 10     |
| Our Sign              | 0.18        | 0.17 | 10     |

## VI. CONCLUSION

This paper proposes a quantum-resilient lattice-based threshold signature framework (QR-LBTS) built upon Crystals-Dilithium and RLWE assumptions, targeting the stringent low-latency and high-security requirements of vehicular networks. Our system innovates in threshold adaptability, consensus-based computation, and protocol-level security reinforcement to realize the reduction of communication overhead and the enhancement of security. By incorporating flexible  $(t, n)$  threshold configurations with adaptive signer selection, neighbor-weighted consensus for local computation and aggregation, and enhanced security mechanisms such as distributed rejection sampling and a share refresh process, QR-LBTS effectively reduces communication overhead and enhances scalability. Experimental results demonstrate that QR-LBTS achieves favorable online signing performance and scalability for threshold-based, security-critical VANET authentication tasks.

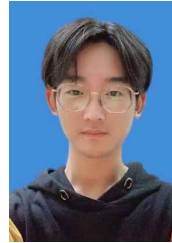
A current limitation of this work is that the VANET-specific modeling is mainly reflected at the authentication and communication level. Richer vehicular features, such as realistic mobility traces, routing dynamics, and cross-layer integration with practical vehicular networking stacks, remain to be incorporated. Accordingly, future work will focus on extending QR-LBTS to more heterogeneous and large-scale vehicular settings, especially under higher mobility, more dynamic topology changes, and less stable communication conditions. We also plan to further investigate implementation-level optimization and real-world deployment issues, including tighter protocol integration with practical vehicular communication systems and more comprehensive evaluation in realistic

WAN environments.

## REFERENCES

- [1] F. Liu and Q. Wang, "Ibirs: An efficient identity-based batch verification scheme for vanets based on ring signature," in *2019 IEEE Vehicular Networking Conference (VNC)*, 2019, pp. 1–8.
- [2] Q. Zhang, "A novel privacy-preserving authentication protocol using bilinear pairing for vanets," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–12, 2021.
- [3] W. Ahmed, W. Di, and D. Mukathe, "A blockchain-enabled incentive trust management with threshold ring signature scheme for traffic event validation in vanets," *Sensors*, vol. 22, no. 17, 2022.
- [4] A. Manasrah, Q. Yaseen, H. Al-Aqrabi, and L. Liu, "Identity-based authentication in vanets: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 4, pp. 4260–4282, 2025.
- [5] A. M. B. I. J. A. Sheeba, and S. Maddu, "Detection of sybil attacks in vanet using transformer with cnn hybrid architecture and optimized by adamw optimizer," in *2025 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI)*, 2025, pp. 1–4.
- [6] I. Amorim and I. Costa, "Leveraging searchable encryption through homomorphic encryption: A comprehensive analysis," *Mathematics*, vol. 11, p. 2948, 2023.
- [7] L. Yan, G. Wang, T. Yin, P. Liu, H. Feng, W. Zhang, H. Hu, and F. Pan, "Attribute-based searchable encryption: A survey," *Electronics*, vol. 13, p. 1621, 2024.
- [8] J. Su and B. Yao, "Random generation topology coding technique in asymmetric topology encryption," *Mathematics*, vol. 12, p. 2768, 2024.
- [9] M. Perepechaenko and R. Kuang, "Quantum encryption of superposition states with quantum permutation pad in ibm quantum computers," *EPJ Quantum Technology*, vol. 10, no. 7, 2023.
- [10] M. Stepniak and J. Mielczarek, "Analysis of multiple overlapping paths algorithms for secure key exchange in large-scale quantum networks," *Journal of Information Security and Applications*, vol. 78, p. 103581, 2023.
- [11] F. Shahid and A. Khan, "Smart digital signatures (sds): A post-quantum digital signature scheme for distributed ledgers," *Future Generation Computer Systems*, vol. 111, pp. 241–253, 2020.
- [12] J. von der Heyden *et al.*, "Privacy-preserving power flow analysis via secure multi-party computation," *IEEE Transactions on Smart Grid*, vol. 16, no. 1, pp. 344–355, 2025.
- [13] V. Sucasas, A. Aly, G. Mantas, J. Rodriguez, and N. Aaraj, "Secure multi-party computation-based privacy-preserving authentication for smart cities," *IEEE Transactions on Cloud Computing*, vol. 11, no. 4, pp. 3555–3572, 2023.
- [14] D. Liu, G. Yu, Z. Zhong, and Y. Song, "Secure multi-party computation with secret sharing for real-time data aggregation in iiot," *Computer Communications*, vol. 224, pp. 159–168, 2024.
- [15] X. Fan, T. Wu, Q. Zheng, Y. Chen, M. Alam, and X. Xiao, "Hse-voting: A secure high-efficiency electronic voting scheme based on homomorphic signcryption," *Future Generation Computer Systems*, vol. 111, pp. 754–762, 2020.
- [16] Y. Lian, X. Huang, and Y. Mu, "Sa3: Self-adaptive anonymous authentication for dynamic authentication policies," *Future Generation Computer Systems*, vol. 30, pp. 133–139, 2014.
- [17] G. Tang, B. Pang, L. Chen, and Z. Zhang, "Efficient lattice-based threshold signatures with functional interchangeability," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 4173–4187, 2023.

- [18] D. Cozzo and N. P. Smart, "Sharing the luov: Threshold post-quantum signatures," in *Proceedings of the IMA International Conference on Cryptography and Coding (IMACC)*, ser. Lecture Notes in Computer Science, vol. 11929. Cham, Switzerland: Springer, 2019, pp. 128–153.
- [19] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.
- [20] A. Boldyreva, "Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme," in *Public Key Cryptography — PKC 2003*, 2002, pp. 31–46.
- [21] H. Xiong, Z. Chen, and F. Li, "Efficient privacy-preserving authentication protocol for vehicular communications with trustworthy," *Security and Communication Networks*, vol. 5, pp. 1441–1451, 2012.
- [22] M. Mohamed, S. M. Islam, and M. Taha, "Efficient vanet safety message delivery and authenticity with privacy preservation," *PeerJ Computer Science*, vol. 7, p. e598, 2021.
- [23] Z. Kang, J. Li, J. Shen, J. Han, Y. Zuo, and Y. Zhang, "Tfs-abs: Traceable and forward-secure attribute-based signature scheme with constant-size," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 9, pp. 9514–9530, 2023.
- [24] W. Susilo, G. Yang, F. Guo, and Q. Huang, "Constant-size ciphertexts in threshold attribute-based encryption without dummy attributes," *Information Sciences*, vol. 429, pp. 349–360, 2018.
- [25] Z. Zhang, W. Huang, Y. Huang, Y. Liao, C. Wu, and S. Zhou, "An attribute-based pre-authenticated secure communication protocol enabling key protection and credential online-upgrading for 5g nr v2x," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 8, pp. 11 325–11 341, 2025.
- [26] R. Shi *et al.*, "Threshold attribute-based credentials with redactable signature," *IEEE Transactions on Services Computing*, vol. 16, no. 5, pp. 3751–3765, 2023.
- [27] L. Zhang, C. Hu, Q. Wu, J. Domingo-Ferrer, and B. Qin, "Privacy-preserving vehicular communication authentication with hierarchical aggregation and fast response," *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2562–2574, 2016.
- [28] J. Duan, S. Zheng, W. Wang, L. Wang, X. Hu, and L. Gu, "Concise ringct protocol based on linkable threshold ring signature," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 5, pp. 5014–5028, 2024.
- [29] J. Wen, L. Bai, Z. Yang, H. Zhang, H. Wang, and D. He, "Larrs: Lattice-based revocable ring signature and its application for vanets," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 1, pp. 739–753, 2024.
- [30] Y. Xie *et al.*, "Accountable and secure threshold eddsa signature and its applications," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 7033–7046, 2024.
- [31] K. Emura, T. Hayashi, and A. Ishida, "Group signatures with time-bound keys revisited: A new model, an efficient construction, and its implementation," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 2, pp. 292–305, 2020.
- [32] C. P. Rentería-Mejía and J. Velasco-Medina, "Lattice-based cryptoprocessor for cca-secure identity-based encryption," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 7, pp. 2331–2344, 2020.
- [33] G. Rossi Figlarz and F. Passuelo Hessel, "Enhancing the 5g-aka protocol with post-quantum digital signature method," in *Advanced Information Networking and Applications. AINA 2024. Lecture Notes on Data Engineering and Communications Technologies*, vol. 202, 2024, pp. 99–110.
- [34] S. Xu, X. Chen, Y. Guo, Y. Yang, S. Wang, S.-M. Yiu, and X. Cheng, "Lattice-based forward secure multi-user authenticated searchable encryption for cloud storage systems," *IEEE Transactions on Computers*, vol. 74, no. 5, pp. 1663–1677, 2025.
- [35] T. Feneuil and M. Rivain, "Threshold computation in the head: Improved framework for post-quantum signatures and zero-knowledge arguments," *Journal of Cryptology*, vol. 38, p. 28, 2025.
- [36] Y. Pan, R. Jin, Y. Liu, and L. Zhang, "Efficient post-quantum attribute-based access control scheme for blockchain-empowered metaverse data management," *Computer Communications*, vol. 234, p. 108092, 2025.
- [37] Y. S. Rao, V. Srivastava, T. Mohanty *et al.*, "Designing quantum-secure attribute-based encryption," *Cluster Computing*, vol. 27, pp. 13 075–13 091, 2024.
- [38] X. Chen, S. Xu, S. Gao, Y. Guo, S. M. Yiu, and B. Xiao, "Fs-llrs: Lattice-based linkable ring signature with forward security for cloud-assisted electronic medical records," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 8875–8891, 2024.
- [39] H. Lu, Y. Zhu, C. E. Chen, and D. Ma, "Toward efficient key extraction of lbc over ring: Fast non-spherical g-lattice sampler and optimized perturbation generation," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 4301–4315, 2024.
- [40] G. Maringer, S. Puchinger, and A. Wachter-Zeh, "Information- and coding-theoretic analysis of the rlwe/mlwe channel," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 549–564, 2023.
- [41] C. Guo, B. Gong, Z. Li, M. Gong, and H. Zhu, "Fort: A forward secure and threshold authorized multi-authority attribute-based signature scheme for multimedia iot," *IEEE Transactions on Multimedia*, vol. 27, pp. 8859–8874, 2025.



**Hao Yuan** (Student Member, IEEE) is currently pursuing the Ph.D. degree with Guizhou University, Guizhou, China. His current research interests include quantum key distribution and lattice-based cryptography in IoV.



**Yuling Chen** received the B.S. degree in mathematics and applied mathematics from Taishan University, Taian, China, in 2006, and the M.S. degree in applied mathematics and the Ph.D. degree in software engineering from Guizhou University, Guiyang, China, in 2009 and 2021, respectively. She is currently a Professor with the State Key Laboratory of Public Big Data, Guizhou University. Her recent research interests include federated learning, blockchain, cryptography and information safety, and artificial intelligence security.



**Jiachun Li** (Member, IEEE) received the B.Sc. degree from Huazhong University of Science and Technology, Wuhan, China, in 2020, and the Ph.D. degree from the NSEC Lab, Shanghai Jiao Tong University, Shanghai, China, in 2025, supervised by Haojin Zhu. He was a Visiting Scholar with Nanyang Technological University, Singapore, from 2023 to 2024, supervised by Jun Luo. He is currently an associate professor with the State Key Laboratory of Public Data, Guizhou University, Guiyang, China. His research interests include embodied intelligence security, smart home security, and metaverse security. He has published papers in top-tier conferences and journals such as USENIX Security, IEEE INFOCOM, IEEE TIFS, IEEE TDSC, and IEEE TMC.



**Jing Sun** is currently pursuing the Ph.D. degree with the State Key Laboratory of Public Big Data, Guizhou University, Guiyang. Her research interests encompass privacy security, image steganography, image encryption and information security.



**Minyi Guo** (Fellow, IEEE) received the BSc and ME degrees in computer science from Nanjing University, China, and a PhD degree in computer science from the University of Tsukuba, Japan. Additionally, he is the Department of Computer Science and Engineering director of the Embedded and Pervasive Computing Center, Shanghai Jiao Tong University. Before joining SJTU, he had been a professor with the School of Computer Science and Engineering, University of Aizu, Japan. He received the national science fund for distinguished young scholars from

NSFC in 2007. His present research interests include parallel/distributed computing, compiler optimizations, Big Data, and cloud computing. He has more than 400 publications in major journals and international conferences in these areas. He received 7 best/highlight paper awards from international conferences, including ALSPOS 2017 and ICCD 2018. He is now editor-in-chief of IEEE Transactions on Sustainable Computing, and on the editorial board of IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Cloud Computing, and Journal of Parallel and Distributed Computing. He is the Zhiyuan chair professor and an ACM distinguished member.