LCMS: Efficient Lattice-based Conditional Privacy-preserving Multi-receiver Signcryption Scheme for Internet of Vehicles

Songshou Dong, Yanqing Yao, Huaxiong Wang, and Yining Liu, *Senior Member*, IEEE

***Abstract*—Internet of Vehicles (IoV) requires robust security and privacy protection mechanisms to enable trusted traffic information exchange, while also requiring low communication and low computing overhead to meet the real-time requirements of IoV. Existing signcryption schemes suffer from quantum vulnerability, inadequate unlinkability/vehicle anonymity, absence of revocability, poor scalability, inadequate management of malicious entities, and high communication and computational overhead. So we propose an efficient lattice-based conditional privacy-preserving multi-receiver signcryption scheme (LCMS) that systematically addresses these gaps through three core innovations: 1) Privacy preservation is achieved via a pseudonym mechanism integrated with certificateless key generation, which ensures vehicle anonymity and weak unlinkability while preventing malicious key generation center and key escrow; 2) Malicious entity management through dynamic revocability and distributed decryption among roadside units, preventing unilateral message access; and 3) Post-quantum efficiency is achieved by leveraging the Learning With Rounding problem to eliminate expensive Gaussian sampling, combined with ciphertext packing techniques. This reduces time overhead, the size of signcryptexts, and communication overhead, while lowering the overall storage overhead of the scheme through the MP12 trapdoor. Security proofs show LCMS achieves Existential Unforgeability under Adaptive Identity Chosen-Message Attack and Indistinguishability under Adaptive Identity Chosen-Ciphertext Attack in the Random Oracle Model, with rigorously validated resistance against multiple IoV-specific attacks. Experimental results via SageMath implementation demonstrate that our scheme exhibits a smaller signcryptext size and lower signcryption/unsigncryption time compared to existing random lattice-based signcryption schemes. Scalability tests with 300 vehicles and 300 roadside units (RSUs) were completed within 230 seconds. Communication overhead analysis confirms practical feasibility for IEEE 802.11p vehicle communication protocol, and RSU serving capability evaluation under realistic vehicle density (100–200/km^2) and speed (40–60 km/h) further validates system practicality. LCMS provides a quantum-resistant, privacy-preserving, and efficient solution for production IoV.**

***Index Terms*—Internet of Vehicles, signcryption, weak unlinkable, certificateless, revocable, multi-receiver, distributed decryption.**

Songshou Dong is with School of Cyber Science and Technology, Beihang University, Beijing 100191, China, Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, Beihang University, Beijing 100083, China, and Division of Mathematical Sciences, Nanyang Technological University, Singapore, 637553 (email: songshou_dong@163.com).

Yanqing Yao is with Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, Beihang University, Beijing 100083, China, and School of Cyber Science and Technology, Beihang University, Beijing 100191, China (email: yaoyq@buaa.edu.cn).

Huaxiong Wang is with Division of Mathematical Sciences, Nanyang Technological University, Singapore, 637553 (e-mail: hxwang@ntu.edu.sg).

Yining Liu is with the School of Data Science and Artificial Intelligence, Wenzhou University of Technology, Wenzhou, 325027, China (Email: lyn7311@sina.com).

(*Corresponding author: Yanqing Yao.*)

I. INTRODUCTION

As road networks become increasingly complex, the frequency of traffic accidents is also increasing. With the rising number of vehicles and the complexity of road conditions, prompt dissemination of road information is crucial [1]. Vehicle-to-everything (V2X) communication and Internet of Vehicles (IoV) are promising approaches to enhance road safety and driving experience [2]. IoV facilitates the exchange of traffic information. The security architecture of this system consists of a key generation center (KGC), traffic management center (TMC), vehicle (equipped with an on-board units (OBU)), and roadside infrastructure. The KGC acts as a central authority overseeing the system, while roadside infrastructure, such as roadside units (RSUs) provide network services and collect traffic data [3]. Vehicles are equipped with OBUs, supporting secure and efficient vehicle-to-vehicle and vehicle-to-infrastructure communications, thereby enabling a smarter driving environment [4].

However, V2X communication typically occurs in open environments, which presents a number of potential risks [5]. The transmitted content is vulnerable to various malicious operations by malicious adversaries. Therefore, it is necessary to ensure the authentication and integrity of the message [6], [7]. Additionally, the movement of a vehicle can be tracked through its sent messages, compromising privacy [8]. Hence, it is important to implement a pseudonym mechanism to anonymize messages, while still allowing the identity of a malicious vehicle to be revealed [8]. Transmitted messages may contain sensitive information, requiring confidentiality measures such as message encryption [9].

Signcryption is a secure approach that ensures authentication, integrity, and confidentiality in information exchange [10]. In particular, the multi-message multi-receiver signcryption scheme is an attractive approach for secure IoV [15], [16], [17], [18], [19]. Currenctly, Signcryption schemes have been widely adopted across various domains [14]-[22], [30], [31].

The majority of current signcryption schemes base their security on large integer factorization, discrete logarithms, elliptic curve cryptosystems (ECC), and bilinear pairings [14]-[19]. However, these are vulnerable to emerging quantum algorithms [11]-[13]. In contrast, lattice-based signcryption schemes offer a promising alternative. Such schemes are

attractive due to their resistance to quantum computing threats, their simplicity and potential for efficiency, and their average-case to worst-case hardness reductions. As a result, there has been a surge in proposals of lattice-based signcryption schemes [20]-[22], [30], [31].

However, current post-quantum secure signcryption schemes still lack the functionalities and efficiency to be practical for IoV. Some schemes have security vulnerabilities, such as the potential for sender identification through public keys or identities, and issues with malicious key generation or decryption. Some schemes are also not efficient enough to support large-scale IoV. Further details on the state of the art are provided in the section on related work.

Energy-efficient self-organizing communication is vital for resource-limited distributed networks. Bio-inspired synchronization and improved oscillator schemes mitigate wireless sensor networks deafness, reduce energy consumption, and enhance reliability [44]-[46]. Existing reviews summarize relevant firefly-based solutions for packet collision and deafness [47]. Such mechanisms have been extended to vehicular networks [48], while study evaluates three machine learning models to secure IoT networks against cyberattacks [49].

A. Our Motivation

In the IoV scenario, there is currently no scheme that can maintain high computational and communication efficiency while resisting malicious KGC, malicious vehicle, and malicious RSU. Therefore, this paper aims to design a signcryption scheme, an efficient lattice-based conditional privacy-preserving multi-receiver signcryption scheme (LCMS) for multiple receivers, which suitable for IoV that maintains high computational and communication efficiency while having high security, features certificateless design, revocability, and distributed decryption capabilities.

B. Our Contribution

We summarize the threefold contribution to this work below.

1) We design our scheme to tackle malicious behavior by the KGC, RSUs, and vehicles in the IoV. The certificateless structure of our scheme means the private keys of vehicles and RSUs are jointly generated by the KGC and the devices. This approach avoids malicious KGC and eliminates key escrow issues. To prevent malicious RSUs from tracking vehicles through vehicular messages, we employ a pseudonym mechanism to conceal the real identity of message senders, allowing each vehicle to apply for multiple pseudonyms, thereby achieving vehicle anonymity and weak unlinkability. Pseudonyms are refreshed after each message transmission to ensure messages cannot be shown to come from the same sender. To prevent vehicles from sending malicious messages, we allow the KGC to reveal the true identity of the senders and revoke their signcryption rights. This ensures their signcryptext will no longer be verified. Additionally, we prevent malicious RSUs from unilaterally decrypting messages by achieving distributed decryption mechanism [35] in our scheme, where signcryptext can be unsigncrypted when all members in some designated group consent to decryption. Besides, we leverage the Learning With Rounding (LWR) problem [26] to minimize expensive discrete Gaussian sampling, along with ciphertext packing techniques [43], to reduce time overhead, signcryptext size, and communication overhead, and reduce the overall storage overhead of our scheme by using MP12 trapdoor.

2) We conduct a formal and informal security analysis of our scheme in the Random Oracle Model (ROM). Our results show that our scheme satisfies Existential Unforgeability under Adaptive Identity Chosen-Message Attack (EUF-aID-CMA); furthermore, by applying the Fujisaki-Okamoto transformation [36], it achieves Indistinguishability under Adaptive Identity Chosen-Ciphertext Attack (IND-aID-CCA2). In addition, we analyze that our scheme can resist various attacks.

3) We implement and benchmark our scheme against other signcryption schemes using the SageMath. Our experiments show that our scheme has significant advantages in both storage overhead and time overhead. Furthermore, a communication overhead analysis shows that our scheme only incurs a small communication overhead at large parameters, showing that our scheme is practical.

C. Outline of This Paper

The paper is organized as follows. Section II provides an overview of related works. Section III introduces notation and the basics of lattice-based cryptography. Section IV defines our LCMS and provides a high-level framework for the scheme, together with security and threat models. Section V contains the full mathematical details of LCMS. Section VI provides correctness analysis and security proofs. Section VII provides performance evaluation and benchmarking. Section VIII gives a summary of our work and some concluding remarks.

II. RELATED WORK

A. (Certificateless) Signcryption Schemes

Signcryption, which combines encryption and signature functionalities, is constructed through three main approaches: public key infrastructure-based, identity-based, and certificateless. In 2008, Barbosa et al. [14] pioneered the concept of a certificateless signcryption scheme. Despite this advancement, the computational cost associated with bilinear pairings remained high in certificateless signcryption scheme. Currently, researchers have increasingly focused on signcryption schemes [15], [16] for multi-receiver scenarios to improve performance. However, these schemes [15], [16] encounter several limitations in IoV scenarios, such as the lack of sender unlinkability, allowing receivers to link senders through fixed public keys; the absence of sender anonymity protection, as they require direct communication using real identities; the lack of necessary traceability and revocation mechanisms [10]; the lack of authentication and integrity security. To resolve these issues, Liang et al. [17] introduced an unlinkable signcryption scheme for multi-receiver IoV. Next, to address the issue of receiver anonymity, Liang et al. [18] proposed a fully anonymous broadcast signcryption scheme for secure health data transmission in Wireless Body Area Networks. However, neither of these schemes is resistant to attacks by quantum computers. Su et al. [19] proposed a

secure and efficient personalized multi-receiver signcryption scheme with cross-domain authentication. This scheme is not post-quantum secure and does have the drawback of ciphertext expansion.

Currently, lattice-based signcryption schemes have seen significant development, addressing various communication and security needs. In 2019, Le et al. [20] proposed a signcryption scheme featuring equality testing within the standard model. Building on these foundations, in 2022, Yu et al. [21] introduced a certificateless multi-source signcryption scheme based on lattices. The following year, they [22] advanced the field with a lattice-based threshold signcryption scheme designed for secure blockchain oracle data transmission. In 2025, Xu et al. [30] proposed an efficient and secure post-quantum certificateless signcryption with linkability for the Internet of Medical Things (IoMT). Bai et al. [31] proposed an efficient module-lattice-based certificateless online/offline signcryption scheme for IoMT. Leveraging module lattice structure, this scheme achieves higher efficiency than traditional lattice-based approaches, but security issues seem to exist. Despite these advances, existing lattice-based signcryption schemes still struggle to fully meet the comprehensive security requirements and efficiency requirements of the IoV, especially in terms of ensuring vehicle unlinkability, vehicle anonymity, revocability, storage overhead, real-time performance, and communication overhead.

III. PRELIMINARIES

This section introduces notation and a basic overview of lattice-based cryptography.

A. Notations

Table I describes some system parameters (the selection of m, $\bar{m}$, w, q, $q^{'}$, σ, σ_s, σ_{lwe}, and σ_{flood} values refer to Ref. [25], [26], [28], [35], [40], and [41]).

TABLE I NOTATIONS AND THEIR DESCRIPTION

Notations	Description
λ	The security parameter
n	The matrix dimension
log	The logarithm based on 2
$\lceil n \rceil$	Round up
$\lfloor n \rfloor$	Round down
$m,\ \bar{m},\ w$	The matrix dimension, $m = 2n\lceil \log q \rceil,\ \bar{m} = w = n\lceil \log q \rceil$
q	The prime modulus $q,\ q \equiv 1 \pmod{2m}$
$q^{'}$	The LWR modulus, $\frac{q}{q^{'}} \geq \sqrt{n}$
$\mathbb{Z}$	The integer field
$\mathbb{Z}_q$	The elements in $\mathbb{Z}_q$ are represented by integers in the range $[-\lfloor \frac{q-1}{2} \rfloor, \lfloor \frac{q-1}{2} \rfloor]$
$\mathbb{Z}_{q^{'}}$	The elements in $\mathbb{Z}_{q^{'}}$ are represented by integers in the range $[-\lfloor \frac{q^{'}-1}{2} \rfloor, \lfloor \frac{q^{'}-1}{2} \rfloor]$
$\mathbb{R}$	The ring $\mathbb{Z}[x]/(x^m+1)$
$\mathbb{R}_q$	The ring $\mathbb{Z}_q[x]/(x^m+1)$
$\boldsymbol{A}_K$	The master public key (MPK), $\boldsymbol{A}_K \in \mathbb{Z}_q^{n \times m}$
$\boldsymbol{R}_K$	The master secret key (MSK), $\boldsymbol{R}_K \in \mathbb{Z}_q^{m \times w}$
$\boldsymbol{A}_S$	The partial public key, $\boldsymbol{A}_S \in \mathbb{Z}_q^{n \times m}$
$\bar{\boldsymbol{A}}$	The revocation public key (RPK), $\bar{\boldsymbol{A}} \in \mathbb{Z}_q^{n \times n}$
r_{ID}	The revocation secret key (RSK), $r_{ID} \in \mathbb{R}_q$
b	The primary revocation secret key (PRSK), $b \in \mathbb{R}_q$
ID	The identity of senders, $ID \in \{0,1\}^*$
PID	The pseudonym of senders, $PID \in \{0,1\}^*$
id	The identity of receivers, $id \in \{0,1\}^*$
$ID(id)$	This represents the sender ID or receiver id
$\boldsymbol{D}_{PID(id)}$	The partial private key (PPK) of the sender PID or the receiver id
$\boldsymbol{C}_{PID(id)}$	The secret value (SV) of the sender PID or the receiver id
$\boldsymbol{P}_{PID(id)}$	The public key of the sender PID or the receiver id
$(\boldsymbol{C}_{PID(id)},\ \boldsymbol{D}_{PID(id)})$	The private key of the sender PID or the receiver id
$\lvert a \rvert$	Absolute value of the real or integer valued scalar a; $\lvert a \rvert = a$ if $a \geq 0$ and $\lvert a \rvert = -a$ if $a < 0$
$\lVert \boldsymbol{a} \rVert_1$	l_1 norm, $\lVert \boldsymbol{a} \rVert_1 = \sum_{i=1}^{n} \lvert a_i \rvert,\ \boldsymbol{a} \in \mathbb{Z}_q^n$
$\lVert \boldsymbol{a} \rVert$	l_2 norm, $\lVert \boldsymbol{a} \rVert = \sqrt{\sum_{i=1}^{n} a_i^2},\ \boldsymbol{a} \in \mathbb{Z}_q^n$
$\lVert \boldsymbol{a} \rVert_\infty$	l_∞ norm, $\lVert \boldsymbol{a} \rVert_\infty = \max(\lvert a_i \rvert),\ \boldsymbol{a} \in \mathbb{Z}_q^n$
$\lVert \boldsymbol{A} \rVert$	The spectral norm of the matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$, $\lVert \boldsymbol{A} \rVert = \sup_{\lVert \boldsymbol{x} \rVert = 1} \lVert \boldsymbol{A}\boldsymbol{x} \rVert$
$O,\ \omega$	The growth of functions
σ	The Short Integer Solution (SIS) Gaussian parameter, $\sigma \geq \lVert \widetilde{\boldsymbol{R}_K} \rVert \times \omega(\sqrt{\log n}) \approx \sqrt{n \log q} \times \omega(\sqrt{\log n})$
κ	Challenge bound, $\binom{n}{\kappa} 2^\kappa \geq 2^{128}$
σ_s	The signature Gaussian parameter, $\sigma_s = \alpha \lVert \boldsymbol{C}\boldsymbol{c} \rVert = 24\sigma\kappa\sqrt{m}$, $\boldsymbol{C}$ is secret key, $\boldsymbol{c}$ is the challenge vector, m is the length of the signature vector
σ_{lwe}	The Learning With Errors (LWE) Gaussian parameter, $\sigma_{lwe} = \frac{q}{2q^{'}\sqrt{3}}$
σ_{flood}	The flooding noise Gaussian parameter, $\sigma_{flood} = \sqrt{2}\sigma_{lwe}$
$\mathcal{R}$	The real numbers field
$\widetilde{\boldsymbol{R}}$	The Gram-Schmidt orthogonalization of matrix $\boldsymbol{R}$
$\boldsymbol{R}^T$	The transpose of the matrix
∂	$\partial: \mathbb{R}_q \to \mathbb{Z}_q^m$ that maps a ring vector $f = f_0 + f_1 \times x + \cdots + f_{m-1} \times x^{m-1} \in \mathbb{R}_q$ to a vector $(f_0, f_1, \cdots, f_{m-1})^T$ over $\mathbb{Z}_q^m$
rot	rot: $\mathbb{R}_q \to \mathbb{Z}_q^{m \times n}$ that maps a ring vector $f = f_0 + f_1 \times x + \cdots + f_{m-1} \times x^{m-1} \in \mathbb{R}_q$ to a matrix $[\partial(f) \mid \partial(f \times x) \mid \cdots \mid \partial(f \times x^{n-1})]$ over $\mathbb{Z}_q^{m \times n}$
S_β	S_β includes all $f \in \mathbb{R}$, s.t. $\lVert \partial(f) \rVert_\infty \leq \beta$
$\otimes$	The Kronecker product
$\oplus$	XOR operation
H_1	The hash function, $H_1: \{0,1\}^* \to \{0,1\}^{\lvert ID \rvert}$, $\lvert ID \rvert$ is bit-length of the binary string ID
H_2	The hash function, $H_2: \{0,1\}^* \to \{\boldsymbol{v}: \boldsymbol{v} \in \{-1,0,1\}^n,\ \lVert \boldsymbol{v} \rVert_1 \leq \kappa\}$
H_3	The hash function, $H_3: \{0,1\}^* \to \mathbb{Z}_q^{n \times n}$
H_4	The hash function, $H_4: \{0,1\}^* \to \{0,1\}^{\lvert \boldsymbol{\varpi} \rvert}$
$\boldsymbol{I}_n$	The $n \times n$ identity matrix
p	The number of registered vehicles
l	The number of the receivers
e	Euler's number, e ≈2.71828
M	The rejection sampling times, $M = \mathrm{e}^{\frac{12}{\alpha} + \frac{1}{2\alpha^2}},\ \alpha = 12$
$\boldsymbol{\varpi}$	The message, $\boldsymbol{\varpi} \in \{0,1\}^*$
$\boldsymbol{id}$	The receivers set, $\boldsymbol{id} = (id_1, \cdots, id_l)$
$\boldsymbol{\varphi}$	The signcryptext
$[n]$	$1, \cdots, n$
$\lfloor n \rceil$	Round to nearest integer

B. Basic Properties and Difficult Assumptions of Lattice

1) Lattice

Some basic knowledges of the lattices are shown in [23]. $\boldsymbol{b}_1,\ \boldsymbol{b}_2, \cdots,\ \boldsymbol{b}_m$ are m linearly independent vectors in $\mathcal{R}^n$. Let $\boldsymbol{B} = [\boldsymbol{b}_1 \mid \boldsymbol{b}_2 \mid \cdots \mid \boldsymbol{b}_m]$, $\Lambda(\boldsymbol{B}) = \{\boldsymbol{B}\boldsymbol{c} = \sum_{i=1}^{m} \boldsymbol{b}_i c_i \mid \boldsymbol{c} \in \mathbb{Z}^m\}$ represent the m-dimensional lattice Λ generated by the basis $\boldsymbol{B}$, where $\boldsymbol{B}$ is a basis of the lattice $\Lambda(\boldsymbol{B})$.

Typically, for a matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$, two special integer lattices modulo q are described as follows:

$$\Lambda_q^{\perp}(\boldsymbol{A}) = \{\boldsymbol{e} \in \mathbb{Z}^m | \boldsymbol{Ae} = 0 \bmod q\}; \quad (1)$$

$$\Lambda_q^{\boldsymbol{u}}(\boldsymbol{A}) = \{\boldsymbol{e} \in \mathbb{Z}^m | \boldsymbol{u} = \boldsymbol{Ae} \bmod q,\ \boldsymbol{u} \in \mathbb{Z}_q^n\}. \quad (2)$$

2) Discrete Gaussian Distribution

Let $n > 0$, $\boldsymbol{c} \in \mathcal{R}^n$ and $\boldsymbol{\Sigma} \in \mathcal{R}^{n\times n}$ be a positive-definite symmetric matrix. The Gaussian function with covariance parameter $\boldsymbol{\Sigma}$ and center parameter $\boldsymbol{c}$ is defined as

$$\rho_{\boldsymbol{\Sigma},\boldsymbol{c}} = \mathrm{e}^{(-\pi(\boldsymbol{x}-\boldsymbol{c})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{c}))}. \quad (3)$$

The Gaussian distribution over the lattice $\Lambda \subseteq \mathrm{span}(\boldsymbol{\Sigma})$ with covariance parameter $\boldsymbol{\Sigma}$ and center parameter $\boldsymbol{c}$ is the distribution with support Λ and probability mass function

$$\mathcal{D}_{\Lambda,\boldsymbol{\Sigma},\mathbf{c}} = \frac{\rho_{\boldsymbol{\Sigma},\boldsymbol{c}}(\boldsymbol{x})}{\sum_{\boldsymbol{y}\in\Lambda}\rho_{\boldsymbol{\Sigma},\boldsymbol{c}}(\boldsymbol{y})}. \quad (4)$$

If $\boldsymbol{\Sigma} = \sigma^2\boldsymbol{I}_n$, we write $\rho_{\sigma,\boldsymbol{c}}$ and $\mathcal{D}_{\Lambda,\sigma,\mathbf{c}}$. We omit $\boldsymbol{c}$ when it is 0. We also define $\mathcal{D}_{\Lambda+\boldsymbol{c},\Sigma} = \mathcal{D}_{\Lambda,\boldsymbol{\Sigma},-\boldsymbol{c}} + \boldsymbol{c}$. For convenience, we let $\rho_{\boldsymbol{\Sigma},\boldsymbol{c}}(S)$ denote the quantity $\sum_{\boldsymbol{y}\in S}\rho_{\boldsymbol{\Sigma},\boldsymbol{c}}(\boldsymbol{y})$ for any countable set S [28].

3) Hard Problems over Lattices

The security of our LCMS relies on the difficult assumptions of the SIS problem [27], [28], the LWE problem [24], and the LWR problem [26].

Definition 1 (SIS problem). Given a matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$, the target of the $SIS_{q,n,m,\beta}$ is to find a nonzero integer vector $\boldsymbol{v} \in \mathbb{Z}_q^m$, which satisfied $\boldsymbol{Av} = \boldsymbol{0}\ mod\ q$ and $\|\tilde{\boldsymbol{v}}\| \le \beta$.

Definition 2 (the Decision version of the SIS problem). Given a pair $(\boldsymbol{A}, \boldsymbol{t})$ decide, with non-negligible advantage, whether it came from the $SIS_{q,n,m,\beta}$ distribution or whether it was generated uniformly at random from $\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^n$.

Definition 3 (the Decision version of the LWE problem). Let n, m, q be positive integers, $\mathcal{D}_\sigma$ be the distribution of the noise $(\sigma \approx \frac{q}{2\sqrt{n}})$. Distinguish between the following two distributions:

Distribution 1: $(\boldsymbol{A}, \boldsymbol{b} = \boldsymbol{As} + \boldsymbol{e})$, where $\boldsymbol{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$, $\boldsymbol{s} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^n,\sigma}$, $\boldsymbol{e} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^m,\sigma}$;

Distribution 2: $(\boldsymbol{A}, \boldsymbol{b})$, where $\boldsymbol{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$, $\boldsymbol{b} \xleftarrow{\$} \mathbb{Z}_q^m$.

Definition 4 (the search version of the LWE problem). Let n, m, q be the positive integers, $\mathcal{D}_\sigma$ be the distribution of the noise $(\sigma \approx \frac{q}{2\sqrt{n}})$. Given $(\boldsymbol{A}, \boldsymbol{b} = \boldsymbol{As} + \boldsymbol{e})$, find $\boldsymbol{s}$, where $\boldsymbol{A} \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$, $\boldsymbol{s} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^n,\sigma}$, $\boldsymbol{e} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^m,\sigma}$.

Definition 5 (LWR problem). The concept of the LWR problem was initially presented by Banerjee and colleagues [26] as a means to enhance the efficiency of pseudorandom generators that are predicated on the LWE problem. A key distinction between the LWE and LWR problems lies in the errors involved. In the LWR problem, errors are not random but are instead deterministic, which has led to the LWR problem being characterized as a deterministic counterpart to the LWE problem. The process of concealing secret data in the LWR framework involves a rounding operation based on a modulus p, diverging from the error introduction method of the LWE problem. The deterministic error in the LWR problem is engineered by reducing the scale from $\mathbb{Z}_q$ to $\mathbb{Z}_p$.

For an n-dimensional vector $\boldsymbol{s}$ over $\mathbb{Z}_q$, the LWR distribution $\mathcal{A}_{n,q,p}^{LWR}(\boldsymbol{s})$ over $\mathbb{Z}_q^n \times \mathbb{Z}_p$ is obtained by choosing a vector $\boldsymbol{a}$ from $\mathbb{Z}_q^n$ uniformly randomly, and returning

$$(\boldsymbol{a}, \left\lfloor \frac{p}{q}(<\boldsymbol{a}, \boldsymbol{s}> mod\ q) \right\rceil) \in \mathbb{Z}_q^n \times \mathbb{Z}_p. \quad (5)$$

As in the LWE problem, $\mathcal{A}_{n,m,q,p}^{LWR}(\boldsymbol{s})$ denotes the distribution of m samples from $\mathcal{A}_{n,q,p}^{LWR}(\boldsymbol{s})$; that is contained in $\mathbb{Z}_q^{m\times n} \times \mathbb{Z}_p^m$. The search LWR problem is defined as finding secret $\boldsymbol{s}$ just as the same way as the search version of the LWE problem. On the contrary, the Decision $LWR_{n,m,q,p}(D)$ problem aims to distinguish the distribution $\mathcal{A}_{n,q,p}^{LWR}(\boldsymbol{s})$ from the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_p$ with m instances for a fixed $\boldsymbol{s} \xleftarrow{\$} D$.

Remark 1: Ref. [40] introduced how to convert an LWR instance to an LWE instance, as follows:

For $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$, $\boldsymbol{s} \in \mathcal{D}_{\mathbb{Z}^m,\sigma}$, we have

$$(\boldsymbol{A}, \left\lfloor \frac{q'}{q}(<\boldsymbol{A}, \boldsymbol{s}> \bmod q) \right\rceil) = (\boldsymbol{A}, \frac{q'}{q}(\boldsymbol{As} \bmod q) + \boldsymbol{e}). \quad (6)$$

We can lift this to a problem modulo q by multiplying by $\frac{q}{q'}$:

$$\frac{q}{q'}\boldsymbol{b} = \boldsymbol{As} + \frac{q}{q'}\boldsymbol{e} \bmod q, \quad (7)$$

where $\frac{q}{q'}\boldsymbol{e}$ is the random variable containing the error introduced by the rounding operation, of which the coefficients are discrete and nearly uniformly distributed in $[-\frac{q}{2q'}, \frac{q}{2q'})$. Ref. [41] introduced how to convert a bounded uniform distribution into a Gaussian distribution, as follows:

For the interval $[-\frac{q}{2q'}, \frac{q}{2q'})$, the transformed Gaussian distribution is $\mathcal{D}_{\mathbb{Z},\sigma}$ $(\sigma = \frac{q}{2q'\sqrt{3}})$.

4) MP12 Trapdoor Functions for Lattices

Ref. [25] gives two polynomial algorithms (GenTrap, SampleD). The details are as follows.

GenTrap(n, m, $\bar{m}$, q): Given n, q, $k = \lceil \log q \rceil$, $\bar{m} = w = n\lceil \log q \rceil$, and $m = \bar{m} + w$, this algorithm samples a matrix $\bar{\boldsymbol{A}} \xleftarrow{\$} \mathbb{Z}_q^{n\times\bar{m}}$, computes $\boldsymbol{g}^T = (1, 2, 4, \cdots, 2^{k-1}) \in \mathbb{Z}^k$ and $\boldsymbol{G} = \boldsymbol{I}_n \otimes \boldsymbol{g}^T$, and outputs a trapdoor $\boldsymbol{R} \in \mathbb{Z}_q^{\bar{m}\times w}$ $(\|\tilde{\boldsymbol{R}}\| \le O(\sqrt{n\log q}))$, and $\boldsymbol{A} = [\bar{\boldsymbol{A}} | \boldsymbol{G} - \bar{\boldsymbol{A}}\boldsymbol{R}] \in \mathbb{Z}_q^{n\times m}$.

SampleD($\boldsymbol{A}$, $\boldsymbol{R}$, $\boldsymbol{u}$, σ): Given the matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$, the vector $\boldsymbol{u} \in \mathbb{Z}_q^n$, the trapdoor $\boldsymbol{R} \in \mathbb{Z}_q^{\bar{m}\times w}$, and the Gaussian parameter $\sigma \ge \|\tilde{\boldsymbol{R}}\| \times \omega(\sqrt{\log n})$, this algorithm outputs $\boldsymbol{x} \in \mathbb{Z}_q^m$ which satisfied $\boldsymbol{Ax} = \boldsymbol{u} \bmod q$ and $\|\tilde{\boldsymbol{x}}\| \le \sigma\sqrt{m}$. According to the algorithm SampleD($\boldsymbol{A}$, $\boldsymbol{R}$, $\boldsymbol{u}$, σ), we can get the algorithm SampleMat($\boldsymbol{A}$, $\boldsymbol{R}$, $\boldsymbol{U}$, σ). The details are as follows.

SampleMat($\boldsymbol{A}$, $\boldsymbol{R}$, $\boldsymbol{U}$, σ): Given the matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$, the matrix $\boldsymbol{U} \in \mathbb{Z}_q^{n\times k}$, the trapdoor $\boldsymbol{R} \in \mathbb{Z}_q^{\bar{m}\times w}$, and the Gaussian parameter $\sigma \ge \|\tilde{\boldsymbol{R}}\| \times \omega(\sqrt{\log n})$, this algorithm outputs $\boldsymbol{X} \in \mathbb{Z}_q^{m\times k}$ which satisfied $\boldsymbol{AX} = \boldsymbol{U} \bmod q$ and $\|\tilde{\boldsymbol{X}}\| \le \sigma\sqrt{m}$.

5) Gaussian Sampling

Lemma 1 [27]. For the discrete Gaussian distribution $\mathcal{D}_{\mathbb{Z}^m,\sigma}$, we have

$$\begin{cases} Pr\left[x \xleftarrow{\$} \mathcal{D}_{\mathbb{Z},\sigma} \mid |x| > 12\sigma\right] < 2^{-100}, m = 1 \\ Pr\left[\boldsymbol{x} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^m,\sigma} \mid \|\boldsymbol{x}\| > 2\sigma\sqrt{m}\right] < 2^{-m}, m > 1 \end{cases} \quad (8)$$

According to [28] and Lemma 1, when $\boldsymbol{c} \in \mathbb{Z}^m$, *and* α *is a positive number, the following conclusions can be drawn:*

$$\begin{cases} Pr\left[\boldsymbol{x} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^m,\sigma} \middle| \frac{\mathcal{D}_{\mathbb{Z}^m,\sigma}(\boldsymbol{x})}{\mathcal{D}_{\mathbb{Z}^m,\sigma,\boldsymbol{c}}(\boldsymbol{x})} = O(1)\right] \\ = 1 - 2^{-\omega(log m)}, \sigma = \omega(\|\boldsymbol{c}\|\sqrt{log m}) \\ Pr\left[\boldsymbol{x} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^m,\sigma} \middle| \frac{\mathcal{D}_{\mathbb{Z}^m,\sigma}(\boldsymbol{x})}{\mathcal{D}_{\mathbb{Z}^m,\sigma,\boldsymbol{c}}(\boldsymbol{x})} < e^{\frac{12}{\alpha}+\frac{1}{2\alpha^2}}\right] \\ = 1 - 2^{-100} \\ , \sigma = \alpha\|\boldsymbol{c}\| \end{cases} \quad (9)$$

6) Rejection Sampling

Given a positive integer m, $\sigma = \omega(\|\boldsymbol{c}\|\sqrt{\log m})$, and an arbitrary set $V \subseteq \mathbb{Z}^m$, $h: V \to \mathrm{R}$ is a distribution whose preimage is V. There is a constant M making the distribution of (1), (2) less than $2^{-\omega(\log m)}/M$ at statistical distance:

(1) $\boldsymbol{c} \xleftarrow{\$} h$, $\boldsymbol{z} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^m,\sigma,\boldsymbol{c}}$, output $(\boldsymbol{z}, \boldsymbol{c})$ with probability $\min(\frac{\mathcal{D}_{\mathbb{Z}^m,\sigma}}{M\mathcal{D}_{\mathbb{Z}^m,\sigma,\boldsymbol{c}}}, 1)$.

(2) $\boldsymbol{c} \xleftarrow{\$} h$, $\boldsymbol{z} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^m,\sigma}$, output $(\boldsymbol{z}, \boldsymbol{v})$ with probability $1/M$.

Moreover, the probability that (1) outputs something is at least $(1 - 2^{-\omega(\log m)})/M$. More concretely, if $\sigma = \alpha\|\boldsymbol{c}\|$ for any positive α, then $M = e^{\frac{12}{\alpha}+\frac{1}{2\alpha^2}}$ $(\alpha = 12)$, the output of algorithm (1) is within statistical distance $2^{-100}/M$ of the output of (2), and the probability that (1) outputs something is at least $(1 - 2^{-100})/M$ [28].

7) Collision Probability

Lemma 2. For any $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ *where* $m = 2n\lceil \log q \rceil$, *for randomly chosen* $\boldsymbol{S} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^{m\times n},\sigma}$ *where* $\sigma \geq \sqrt{n \log q} \times \omega(\sqrt{\log n})$, *with probability* $1 - 2^{-100}$, *there exists another* $\boldsymbol{S}' \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^{m\times n},\sigma}$ *such that* $\boldsymbol{AS} = \boldsymbol{AS}'$.

Proof: Notice that $\boldsymbol{A}$ *can be thought of as a linear transformation whose range has size* $q^{n\times n}$. *This means that there are at most* $q^{n\times n}$ *elements* $\boldsymbol{S} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^{m\times n},\sigma}$ *that do not collide with any other element in* $\mathcal{D}_{\mathbb{Z}^{m\times n},\sigma}$. *Since the set* $\mathcal{D}_{\mathbb{Z}^{m\times n},\sigma}$ *consists of* $\approx (2\sigma)^{n\times m}$ *elements, the probability of sampling a non-colliding element is at most*

$$\frac{q^{n\times n}}{(2\sigma)^{n\times m}} \leq \left(\frac{q^n}{(2O(\sqrt{n\log q}) \times \omega(\sqrt{\log n}))^{2n\lceil \log q \rceil}}\right)^n < 2^{-100}. \quad (10)$$

IV. FRAMEWORK OF OUR LCMS

In this section, the primary framework of our LCMS is presented, encompassing the system model, the definition of LCMS, the security model, the threat model, and the design requirements.

A. System Model

Our LCMS consists of four main components: KGC, the vehicle, the RSU, and TMC. Fig. 1 shows our system model.

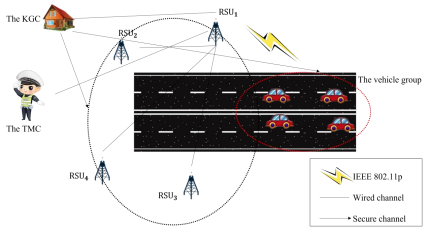


Fig. 1. System model.

In Fig. 1, lightning bolt indicate the IEEE 802.11p vehicle communication protocol [42], a wireless communication standard specifically used for short-range V2X roadside communication, which marks the interactions between vehicles and the smallest-*id* RSU in the group. This aligns with IEEE 802.11p's application scenario for short-range V2X roadside communication. Solid black lines represent wired communication, which applies to data transmission between the KGC and all RSUs (ensuring stable, high-bandwidth connections for system-level management), between the TMC and all RSUs, and between RSUs (especially for the smallest-id RSU in the group to send signcryptexts to other RSUs in the group and receive their processed results). Solid black lines with arrows denote secure channels for encrypted interactions, which includes: 1) the KGC and RSUs, for credential issuance; and 2) the KGC and vehicles, for secure registration and credential issuance.

First, the registered vehicle group sends the signcryptext to the smallest-*id* RSU in the group via IEEE 802.11p vehicle communication protocol [42]. Upon receiving the signcryptext, the smallest-*id* RSU in the group sends it to other RSUs in the group through a wired channel. These RSUs then process the signcryptext and transmit their results back to the smallest-*id* RSU in the group. The smallest-*id* RSU in the group then unsigncrypts the data to obtain the plaintext and sends it to other RSUs in the group through a wired channel, and only after unsigncryption can it identify malicious content in the message. If malicious content is detected, the smallest-*id* RSU in the group signs the pseudonym with its private key, and sends it to the KGC via wired communication. The KGC can verify the signature, reveal the vehicle's true identity, and revoke its signcryption rights.

1) The KGC: Each region in the system has one KGC, the only entity with strong computing and storage capabilities. It is responsible for system initialization, generating the PPK for vehicles and RSUs, the RSKs for vehicles, and pseudonyms for vehicles. Additionally, it handles tracking and revocation of malicious vehicles (based on RSU reports) and remains a trusted entity in the system.

2) The vehicles: Each vehicle must complete secure registration with the KGC and be equipped with tamper-resistant devices and OBUs. These OBUs store confidential information and generate the signcryptext. In normal operation, vehicles send data directly to RSUs. However, vehicles may act as malicious entities (e.g., forged signatures or malicious data), so their signcrypted data still requires RSU verification and KGC supervision.

3) The RSU: As a key roadside infrastructure node in each KGC-managed region, it mainly interacts with the vehicles, KGC, and TMC. Its core tasks include three main functions. First, it receives the signcryptext transmitted by the vehicles.

Second, it unsigncrypts the signcryptext when all RSUs in the designated group reach a consensus, ensuring data security, and providing traffic services to nearby vehicles. Third, it forwards traffic feedback to the TMC. Additionally, if an RSU detects potential malicious behavior (e.g., abnormal vehicle data), it can submit evidence to the KGC for handling. Note that the RSU itself may act as a malicious entity in the data interaction chain.

4) The TMC: As the upper-level traffic management entity, it maintains secure wired connections with all RSUs (not vehicles). Its core functions include: aggregating traffic data from RSUs for real-time monitoring and route planning; coordinating emergency directives with the KGC; and auditing RSU operations. The TMC operates as a fully trusted entity, interacting with KGC for cross-system coordination and with RSUs for data collection.

Remark 2: The system model describes a per-region KGC deployment, reflecting real-world traffic management boundaries. However, the cryptographic construction and security proofs in sections V-VI are scoped to a single KGC domain. This design choice is intentional, as V2X safety messages are predominantly intra-regional and require minimal latency. Cross-region communication can be handled at the application layer through KGC-to-KGC secure channels. This approach does not alter the core signcryption primitive. Extending our scheme to a fully federated multi-KGC architecture remains an interesting direction for future research.

B. Definition of LCMS

Our LCMS encompasses the following nine algorithms. The workflow of LCMS is described in Fig. 2.

1) Setup: It is run by the KGC. Given the security parameter λ, it returns the public parameters, the MPK $\boldsymbol{A}_K$, and the MSK $\boldsymbol{R}_K$.

2) *Extract the pseudonym:* It is run by the KGC. Given a vehicle's real identity ID, it returns its pseudonym PID.

3) *Set the public key:* It is run by the vehicle (or RSU). Given a vehicle's pseudonym (or RSU's identity), it returns the public key $\boldsymbol{P}_{PID(id)}$ and its identity $PID(id)$.

4) *Set the PPK:* It is run by the KGC. Given a vehicle's pseudonym (or RSU's identity), its public key $\boldsymbol{P}_{PID(id)}$, and the MSK $\boldsymbol{R}_K$, it returns PPK $\boldsymbol{D}_{PID(id)}$.

5) *Set the private key:* It is run by the vehicle (or RSU). Given PPK $\boldsymbol{D}_{PID(id)}$, it returns the private key ($\boldsymbol{C}_{PID(id)}$, $\boldsymbol{D}_{PID(id)}$).

6) *Join:* It is run by the KGC. Given the number p of vehicles, it returns p-many RSKs r_{ID_i}, the revocation polynomial (RP) $f(x)$, and RPK $\overline{\boldsymbol{A}}$.

7) *Signcryption:* It is run by a vehicle. Given the vehicle's pseudonym PID_s, the message $\boldsymbol{\varpi}$, the vehicle' private key sk_{PID_s}, the vehicle's RSK r_{ID_s}, a set of RSUs' identities $\boldsymbol{id} = (id_1, \cdots, id_l)$, the RSUs' public key set pk_{id_i} $(i \in [l])$, and the current timestamp $\boldsymbol{t}_{PID_s}$, it returns $\boldsymbol{\varphi}_{PID_s} = (\boldsymbol{t}_{PID_s}, \boldsymbol{z}_{PID_s,1}, \boldsymbol{z}_{PID_s,2}, \boldsymbol{c}_{PID_s}, \boldsymbol{x}, \boldsymbol{c}_{\boldsymbol{id},1}, \boldsymbol{c}_{\boldsymbol{id},2}, \boldsymbol{c}_{\boldsymbol{id},3}, \boldsymbol{id})$ as the corresponding signcryptext.

8) *Unsigncryption:* It is run by a set of RSUs. Given the signcryptext $\boldsymbol{\varphi}_{PID_s}$, the vehicle' public key pk_{PID_s}, the RSUs' private key set sk_{id_i} $(i \in [l])$, it returns the message $\boldsymbol{\varpi}$ or $\perp$.

9) *Trace and revocation:* It is run by the KGC. Given the pseudonym PID of malicious vehicle, it restores the real identity ID of malicious vehicle and revokes the signcryption qualification of malicious vehicle.

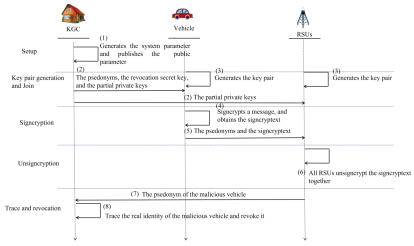


Fig. 2. Workflow chart.

C. Security Model

According to Ref. [14], two types of adversaries emerge in the certificateless signcryption scheme. $\mathcal{A}_1$ only can replace the public key, while $\mathcal{A}_2$ only can get the MSK. In this case, our LCMS should ensure confidentiality and unforgeability of the signcryptext. In other words, the EUF-aID-CMA and the IND-aID-CCA2 properties from $\mathcal{A}_1$ and $\mathcal{A}_2$ should be satisfied in our scheme. Details are shown in Table II.

TABLE II SECURITY OF OUR SCHEME IN THE ROM

Security	Unforgeability (EUF-aID-CMA)	Confidentiality (IND-aID-CCA2)
Game	Game I	Game II
Detailed process	*Setup:* Give the security parameter λ, challenger $\mathcal{C}$ obtains public parameters and the MSK $\boldsymbol{R}_K$, keeps the MSK $\boldsymbol{R}_K$ secret, sends public parameters to $\mathcal{A}_1$ or $\mathcal{A}_2$, and sends MSK $\boldsymbol{R}_K$ to $\mathcal{A}_2$.	
	Query phase 1: $\mathcal{A}_1$ or $\mathcal{A}_2$ executes the oracle queries, and $\mathcal{C}$ returns the corresponding responses. *Remark 3:* $\mathcal{A}_1$ or $\mathcal{A}_2$ can execute the following queries: RSK query, hash query, PPK query, SV query, signcryptext query, and unsigncryption query (just for IND-aID-CCA2). The difference appears in the SV query, $\mathcal{A}_1$ may replace the user's public key, whereas $\mathcal{A}_2$ cannot replace the user's public key, but is given the MSK.	
	Forgery: $\mathcal{A}_1$ or $\mathcal{A}_2$ outputs a forged signcryptext $\boldsymbol{\varphi}_{PID^*}$ on (ID^*, PID^*), $\boldsymbol{\varpi}^*$, $\boldsymbol{t}_{PID^*}$, and $\boldsymbol{id}^* = (id_1^*, id_2^*, \cdots, id_l^*)$. If $\boldsymbol{\varphi}_{PID^*}$ can be unsigncrypted by any receiver, (ID^*, PID^*) had not been queried by $\mathcal{A}_1$ for the PPK query, (ID^*, PID^*) had not been queried by $\mathcal{A}_2$ for the SV query, and (PID^*, $\boldsymbol{\varpi}^*$, $\boldsymbol{t}_{PID^*}$, $\boldsymbol{id}^*$) had not been queried by $\mathcal{A}_1$ or $\mathcal{A}_2$ for the signcryptext query, $\mathcal{A}_1$ or $\mathcal{A}_2$ wins the game.	*Challenge:* $\mathcal{A}_1$ or $\mathcal{A}_2$ samples two messages ($\boldsymbol{\varpi}_0$, $\boldsymbol{\varpi}_1$), the timestamp $\boldsymbol{t}_{PID^*}$, and the identities (ID^*, PID^*) and $\boldsymbol{id}^* = (id_1^*, id_2^*, \cdots, id_l^*)$. And $\mathcal{A}_1$ or $\mathcal{A}_2$ submits them to $\mathcal{C}$. Next, $\mathcal{C}$ samples a $c \xleftarrow{\$} \{0, 1\}$ and executes $\boldsymbol{\varphi}_{PID^*} = \text{Signcryption}(ID^*, PID^*, \boldsymbol{\varpi}_c, \boldsymbol{t}_{PID^*}, \boldsymbol{id}^*)$. Finally, $\mathcal{C}$ sends $\boldsymbol{\varphi}_{PID^*}$ to $\mathcal{A}_1$ or $\mathcal{A}_2$.
		Query phase 2: Except for the PPK query about $\boldsymbol{id}^*$, which cannot be queried by adversary $\mathcal{A}_1$, the SV query about $\boldsymbol{id}^*$ cannot be queried by adversary $\mathcal{A}_2$, and the unsigncryption query about $\boldsymbol{\varphi}_{PID^*}$ cannot be queried by adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$, other queries are the same as the queries in Query phase 1.
		Guess: $\mathcal{A}_1$ or $\mathcal{A}_2$ returns a coin $c' \in \{0, 1\}$. And $\mathcal{A}_1$ or $\mathcal{A}_2$ win the game if $c' = c$, $\boldsymbol{id}^*$ had not been queried by $\mathcal{A}_1$ for the PPK query, and $\boldsymbol{id}^*$ had not been queried by $\mathcal{A}_2$ for the SV

		query.
Probability advantage	$Adv_{\mathrm{LCMS}}^{\mathrm{EUF-aID-CMA}}(\mathcal{A})$ $= \Pr[\mathrm{Unsigncryption}(PID^*, \boldsymbol{\varphi}_{PID^*}, \boldsymbol{id}^*) \neq \perp]$	$Adv_{\mathrm{LCMS}}^{\mathrm{IND-aID-CCA2}}(\mathcal{A}) = \left\lvert \Pr[c' - c] - 1/2 \right\rvert$

D. Threat Model

We adopt a layered threat model that distinguishes system trust assumptions from adversarial capabilities in formal proofs. In the system model (Section IV-A), the KGC and TMC are trusted for operations; however, our formal security model (Section IV-C) considers stronger adversaries, including a compromised KGC. This is standard in certificateless cryptography [14]: the scheme remains secure even if the MSK is leaked, provided the adversary cannot replace public keys.

Adversary Classes:

1) External eavesdroppers (fully malicious). They control the wireless channel, eavesdrop on V2X communications, inject forged signcryptexts, and perform traffic analysis. These are mitigated by IND-aID-CCA2, EUF-aID-CMA, and the pseudonym refresh mechanism, respectively.

2) Vehicles (malicious). Malicious vehicles actively deviate from the protocol to: (1) inject false data, (2) launch replay attacks, and (3) bypass revocation. These are mitigated by revocation, timestamp validation, and RP-based RSK updates, respectively.

3) RSUs (malicious). Malicious RSUs can actively attempt to track vehicles by correlating pseudonyms (prevented by pseudonym refresh), unilateral decryption (prevented by distributed decryption), and modify messages (detected by verification algorithm).

4) Type I adversary $\mathcal{A}_1$ (formal model). An external adversary that does not possess the MSK but can replace public keys of users. It captures malicious users or compromised infrastructure [14].

5) Type II Adversary $\mathcal{A}_2$ (Formal model). An internal adversary that possesses the MSK but cannot replace public keys of users. It captures insider threats (e.g., MSK leakage), and our proofs demonstrate that the scheme remains secure against such adversaries [14].

Alignment with formal proofs. The proofs in Section VI-B consider both $\mathcal{A}_1$ and $\mathcal{A}_2$ under the ROM. EUF-aID-CMA holds against both types, and IND-aID-CCA2 holds under the Decision-SIS assumption, guaranteeing security against both external attackers and insider threats.

E. Design Requirements

To ensure the secure and efficient operation of our scheme, the following objectives must be met concerning data reliability, privacy for legitimate participants, and the management of adversarial entities:

1) Data security and trustworthiness: The transmitted data must satisfy authentication, integrity, and confidentiality. This involves:

Authentication and integrity: Verifying the legitimacy of the transmitting vehicle and safeguarding the data against unauthorized modifications during transit.

Confidentiality: Sensitive information must be encrypted into signcryptext before transmission over public channels, thereby protecting it from eavesdropping.

2) Privacy preservation for authorized entities: This is achieved through:

Vehicle anonymity: Guaranteeing that only the KGC can ascertain a vehicle's true identity from its pseudonym, keeping it anonymous to all other entities.

Weak unlinkability: By alternating between different pseudonyms, we ensure that no entity can infer the correlation between two consecutive messages.

3) Response to the malicious entities: The system must promptly identify and neutralize vehicles exhibiting malicious behavior through:

Traceability: Allow the KGC to trace and verify the genuine identity of any vehicle involved in a dispute.

Revocability: Empower the KGC to effectively remove a malicious vehicle from the system after it has been identified.

4) Distributed decryption mechanism: The signcryptext can be decrypted only if a specific group of recipients all agree to decrypt it.

5) Resistance to various security threats: Our LCMS is designed to withstand a spectrum of attacks, including but not limited to:

Protection against impersonation, modification, replay, eavesdropping, man-in-the-middle attacks and malicious KGC attacks.

V. OUR SCHEME

In this section, we will present the LCMS, which contains nine algorithms: Setup, Extract the pseudonym, Set the public key, Set the PPK, Set the private key, Join, Signcryption, Unsigncryption, and Trace and revocation. Next are the detailed descriptions of the nine algorithms. Besides, in our scheme, pseudonym mechanism ensures the weak unlinkability of the vehicle, the lattice structure ensures the security of scheme in the post quantum era, the certificateless structure avoids malicious KGC and key escrow, the using of LWR problem [26] reduces expensive Gaussian sampling, along with ciphertext packing techniques [43], to reduce time overhead, signcryptext size, and communication burden, the MP12 reduces the overall storage overhead of our scheme, the distributed decryption mechanism [35] strengthens the security of the scheme. The parameter selection of our scheme refers to Ref. [25], [26], [28], [35], [40], and [41].

1) Setup: Given the security parameter λ, the KGC needs to prepare the following parameters:

The matrix's dimension n, the prime modulus q ($q \equiv 1 \pmod{2m}$), the LWR modulus q' ($\frac{q}{q'} \geq \sqrt{n}$), $m = 2n\lceil \log q \rceil$, $\bar{m} = w = n\lceil \log q \rceil$, the challenge bound κ ($\binom{n}{\kappa} 2^{\kappa} \geq 2^{128}$), the SIS Gaussian parameter $\sigma \geq \sqrt{n \log q} \times \omega(\sqrt{\log n})$, the signature Gaussian parameter $\sigma_s = \alpha \|\boldsymbol{Cc}\| = 24\sigma\kappa\sqrt{m}$, the LWE Gaussian parameter $\sigma_{lwe} = \frac{q}{2q'\sqrt{3}}$, and the flooding noise Gaussian parameter $\sigma_{flood} = \sqrt{2}\sigma_{lwe}$.

Next, the KGC defines the anti-collision hash functions $H_1: \{0,1\}^* \to \{0,1\}^{|ID|}$, $H_2: \{0,1\}^* \to \{\boldsymbol{v}: \boldsymbol{v} \in \{-1,0,1\}^n$,

$\|\boldsymbol{v}\|_1 \le \kappa\}$, $H_3: \{0,1\}^* \to \mathbb{Z}_q^{n\times n}$, and $H_4: \{0,1\}^* \to \{0,1\}^{|\boldsymbol{\varpi}|}$, samples the matrix $\boldsymbol{A}_S \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, and computes $(\boldsymbol{A}_K, \boldsymbol{R}_K) \leftarrow \mathrm{GenTrap}(n, m, \bar{m}, q)$, where the MPK is $\boldsymbol{A}_K \in \mathbb{Z}_q^{n\times m}$, and MSK is $\boldsymbol{R}_K$ ($\|\widetilde{\boldsymbol{R}_K}\| \le O(\sqrt{n\log q})$).

2) Extract the pseudonym: Given the vehicle's identity ID_i, KGC samples $PID_i^1 \in \{0,1\}^*$, computes $PID_i^2 = H_1(\boldsymbol{R}_K, PID_i^1) \oplus ID_i$ and sends $PID_i = (PID_i^1, PID_i^2)$ to the vehicle ID_i (to maintain weak unlinkability, vehicle ID_i can apply for multiple pseudonyms.).

3) Set the public key: Vehicle (or RSU) $PID_i(id_i)$ samples the matrix $\boldsymbol{C}_{PID_i(id_i)} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^{m\times n},\sigma}$, computes $\boldsymbol{A}_S\boldsymbol{C}_{PID_i(id_i)} = \boldsymbol{P}_{PID_i(id_i)} \bmod q$, and sends $PID_i(id_i)$ to the KGC.

4) Set the PPK: On reception $PID_i(id_i)$, KGC computes $\boldsymbol{H}_{PID_i(id_i)} = H_3(PID_i(id_i))$, and $\boldsymbol{D}_{PID_i(id_i)} \leftarrow \mathrm{SampleMat}(\boldsymbol{A}_K, \boldsymbol{R}_K, \boldsymbol{H}_{PID_i(id_i)}, \sigma)$, where $\boldsymbol{A}_T\boldsymbol{D}_{PID_i(id_i)} = \boldsymbol{H}_{PID_i(id_i)} \bmod q$, $\boldsymbol{D}_{PID_i(id_i)} \in \mathbb{Z}^{m\times n}$, $\|\widetilde{\boldsymbol{D}_{PID_i(id_i)}}\| \le \sigma\sqrt{m}$. Finally, KGC sends $\boldsymbol{D}_{PID_i(id_i)}$ to the vehicle (or RSU) $PID_i(id_i)$ by the secure channel.

5) Set the private key: On reception $\boldsymbol{D}_{PID_i(id_i)}$, the vehicle (or RSU) computes as follows:

(1) First, calculate $\boldsymbol{H}_{PID_i(id_i)} = H_3(PID_i(id_i))$, check whether $\boldsymbol{A}_K\boldsymbol{D}_{PID_i(id_i)} = \boldsymbol{H}_{PID_i(id_i)} \bmod q$ holds and $\|\widetilde{\boldsymbol{D}_{PID_i(id_i)}}\| \le \sigma\sqrt{m}$. If so, continue with the next step; otherwise, terminate the process.

(2) Set $pk_{PID_i(id_i)} = \boldsymbol{P}_{PID_i(id_i)}$, $sk_{PID_i(id_i)} = (\boldsymbol{C}_{PID_i(id_i)}, \boldsymbol{D}_{PID_i(id_i)})$ (PID_i ($i \in [p]$)). The key generation process is shown in Fig. 3.

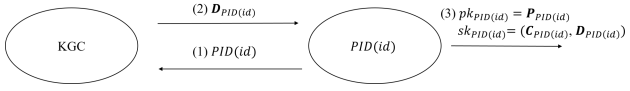


Fig. 3. Key generation process.

6) Join: Given the number p of vehicles, the KGC samples p RSKs $r_{ID_i} \xleftarrow{\$} \mathbb{R}_q$ and the PRSK $b \xleftarrow{\$} S_1$, computes $\boldsymbol{A}_K\boldsymbol{B} = \bar{\boldsymbol{A}} \bmod q$ ($\boldsymbol{B} = \mathrm{rot}(b)$) and $f(x) = \prod_{i=1}^{p}(x - r_{ID_i}) + b$, and publishes $f(x)$ as the RP and $\bar{\boldsymbol{A}}$ as the RPK. Finally, KGC sends r_{ID_i} to each vehicle ID_i separately by the secure channel. Fig. 4 shows the distribution process of the RSK.

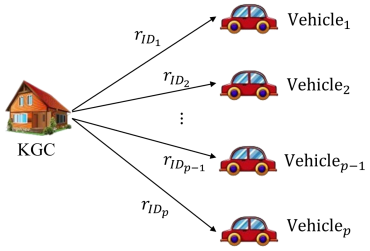


Fig. 4. Distribution process of RSKs.

Remark 4: Even if a malicious signer attempts to recover the RSK of another user, the best-known approach requires a time complexity of $O(mq + p^m m\log m)$. Thus, computationally infeasible for any realistic adversary.

7) Signcryption: Given vehicle' pseudonym PID_s, the private key $sk_{PID_s} = (\boldsymbol{C}_{PID_s}, \boldsymbol{D}_{PID_s})$, the RSK r_{ID_s}, message $\boldsymbol{\varpi} \in \{0,1\}^*$, RSU identities set $\boldsymbol{id} = (id_1, \cdots, id_l)$ and its public key set $pk_{id_i} = \boldsymbol{P}_{id_i}$ ($i \in [l]$). The vehicle PID_s computes as follows:

(1) Sample the current timestamp $\boldsymbol{t}_{PID_s} \xleftarrow{\$} \{0,1\}^*$, $\boldsymbol{y}_{PID_s,1} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^m,\sigma_s}$ and $\boldsymbol{y}_{PID_s,2} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^m,\sigma_s}$, and computes $\boldsymbol{c}_{PID_s} = H_2\left(\begin{bmatrix}\boldsymbol{A}_K\boldsymbol{y}_{PID_s,1}\\ \boldsymbol{A}_S\boldsymbol{y}_{PID_s,2}\end{bmatrix} \bmod q, \boldsymbol{\varpi}_{PID_s}, \boldsymbol{t}_{PID_s}\right)$.

(2) Compute $\boldsymbol{z}_{PID_s,1} = (\boldsymbol{D}_{PID_s} + \boldsymbol{B})\boldsymbol{c}_{PID_s} + \boldsymbol{y}_{PID_s,1}$ ($\|(\boldsymbol{D}_{PID_s} + \boldsymbol{B})\boldsymbol{c}_{PID_s}\| \ne \boldsymbol{0}$, $\boldsymbol{B} = \mathrm{rot}\left(f\left(r_{ID_s}\right)\right)$) and $\boldsymbol{z}_{PID_s,2} = \boldsymbol{C}_{PID_s}\boldsymbol{c}_{PID_s} + \boldsymbol{y}_{PID_s,2}$ ($\|\boldsymbol{C}_{PID_s}\boldsymbol{c}_{PID_s}\| \ne \boldsymbol{0}$).

(3) The vehicle PID_s outputs signature ($\boldsymbol{z}_{PID_s,1}$, $\boldsymbol{z}_{PID_s,2}$, $\boldsymbol{c}_{PID_s}$) with the probability of $\min\left(1, \frac{\mathcal{D}_{\mathbb{Z}^m,\sigma_s}(\boldsymbol{z}_{PID_s,1})}{M_1\mathcal{D}_{\mathbb{Z}^m,\sigma_s,(\boldsymbol{D}_{PID_s}+\boldsymbol{B})\boldsymbol{c}_{PID_s}}(\boldsymbol{z}_{PID_s,1})}, \frac{\mathcal{D}_{\mathbb{Z}^m,\sigma_s}(\boldsymbol{z}_{PID_s,2})}{M_2\mathcal{D}_{\mathbb{Z}^m,\sigma_s,\boldsymbol{C}_{PID_s}\boldsymbol{c}_{PID_s}}(\boldsymbol{z}_{PID_s,2})}\right)$ ($M_1 \approx 3$, $M_2 \approx 3$, $M = \max(M_1, M_2)$).

(4) Sample $\boldsymbol{\tau} \xleftarrow{\$} \{0,1\}^n$ and $\boldsymbol{r} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^n,\sigma_{lwe}}$, compute $\boldsymbol{x} = \boldsymbol{\varpi} \oplus H_5(\boldsymbol{\tau})$, $\bar{\boldsymbol{P}}_{\boldsymbol{id}} = \sum_{id_i \in \boldsymbol{id}}(H_3(id_i) + \boldsymbol{P}_{id_i}) \bmod q$, $\boldsymbol{c}_{\boldsymbol{id},1} = \left\lfloor \frac{q'}{q} \times \left(\boldsymbol{A}_K{}^T\boldsymbol{r} \bmod q\right)\right\rceil$, $\boldsymbol{c}_{\boldsymbol{id},2} = \left\lfloor \frac{q'}{q} \times \left(\boldsymbol{A}_S{}^T\boldsymbol{r} \bmod q\right)\right\rceil$, and $\boldsymbol{c}_{\boldsymbol{id},3} = \left\lfloor \frac{q'}{q} \times \left(\bar{\boldsymbol{P}}_{\boldsymbol{id}}{}^T\boldsymbol{r} \bmod q\right)\right\rceil + \left\lfloor \frac{q'}{2}\right\rfloor \boldsymbol{\tau} \bmod q'$, and send $\boldsymbol{\varphi}_{PID_s} = (\boldsymbol{t}_{PID_s}, PID_s, \boldsymbol{z}_{PID_s,1}, \boldsymbol{z}_{PID_s,2}, \boldsymbol{c}_{PID_s}, \boldsymbol{x}, \boldsymbol{c}_{\boldsymbol{id},1}, \boldsymbol{c}_{\boldsymbol{id},2}, \boldsymbol{c}_{\boldsymbol{id},3}, \boldsymbol{id})$ to the RSU id_1.

8) Unsigncryption: Given the signcryptext $\boldsymbol{\varphi}_{PID_s} = (\boldsymbol{t}_{PID_s}, PID_s, \boldsymbol{z}_{PID_s,1}, \boldsymbol{z}_{PID_s,2}, \boldsymbol{c}_{PID_s}, \boldsymbol{x}, \boldsymbol{c}_{\boldsymbol{id},1}, \boldsymbol{c}_{\boldsymbol{id},2}, \boldsymbol{c}_{\boldsymbol{id},3}, \boldsymbol{id})$ matching the public key set $\boldsymbol{P}_{PID_s}$ and $\boldsymbol{P}_{id_i}$ ($i \in [l]$), RSU id_1 sends $(\boldsymbol{c}_{\boldsymbol{id},1}, \boldsymbol{c}_{\boldsymbol{id},2})$ to other RSUs in $\boldsymbol{id}$. The subsequent decryption steps are as follows:

(1) RSU id_i ($i \in [l]$) samples $\boldsymbol{e}_{id_i,1} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^n,\sigma_{flood}}$ and $\boldsymbol{e}_{id_i,2} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^n,\sigma_{flood}}$, and calculates $\boldsymbol{c}_{id_i,1} = \boldsymbol{D}_{id_i}{}^T\boldsymbol{c}_{\boldsymbol{id},1} + \boldsymbol{e}_{id_i,1} \bmod q'$ and $\boldsymbol{c}_{id_i,2} = \boldsymbol{C}_{id_i}{}^T\boldsymbol{c}_{\boldsymbol{id},2} + \boldsymbol{e}_{id_i,2} \bmod q'$.

(2) Each RSU in $\boldsymbol{id}$ sends its $(\boldsymbol{c}_{id_i,1}, \boldsymbol{c}_{id_i,2})$ to the RSU id_1 (here, as long as one RSU disagrees with decryption in the group, the plaintext cannot be obtained). Finally, the RSU id_1 calculates $\boldsymbol{c}_1 = \sum_{id_i \in \boldsymbol{id}} \boldsymbol{c}_{id_i,1} \bmod q'$, $\boldsymbol{c}_2 = \sum_{id_i \in \boldsymbol{id}} \boldsymbol{c}_{id_i,2} \bmod q'$, and $\hat{\boldsymbol{\tau}} = \boldsymbol{c}_{\boldsymbol{id},3} - \boldsymbol{c}_1 - \boldsymbol{c}_2 \bmod q'$. Let $\hat{\boldsymbol{\tau}} = (\tau_1', \tau_2', \cdots, \tau_n')$, if $\tau_i' \in [-\lfloor \frac{q'}{4}\rfloor, \lfloor \frac{q'}{4}\rfloor)$, let $\tau_i = 0$, otherwise $\tau_i = 1$. Then $\boldsymbol{\tau} = (\tau_1, \tau_2, \cdots, \tau_n)$.

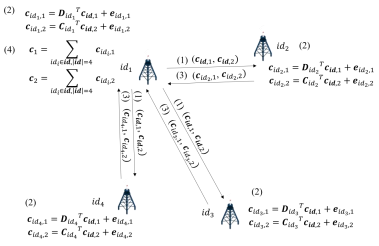


Fig. 5. Distributed decryption process.

(3) RSU id_1 computes $\boldsymbol{\varpi} = \boldsymbol{x} \oplus H_4(\boldsymbol{\tau})$. Fig. 5 shows the process of distributed decryption.

Next, if the following relationship holds, the signcryptext is legal; otherwise, the signcryptext is not legitimate (this verification only needs to be done once by RSU id_1).

(1) $\|\boldsymbol{z}_{PID_s,1}\| \le 2\sigma_s\sqrt{m}$ and $\|\boldsymbol{z}_{PID_s,2}\| \le 2\sigma_s\sqrt{m}$;

(2) $\boldsymbol{c}_{PID_s} = H_2\left(\begin{bmatrix}\boldsymbol{A}_K\boldsymbol{z}_{PID_s,1} - (H_3(PID_s)+\overline{\boldsymbol{A}})\boldsymbol{c}_{PID_s}\\ \boldsymbol{A}_S\boldsymbol{z}_{PID_s,2} - \boldsymbol{P}_{PID_s}\boldsymbol{c}_{PID_s}\end{bmatrix} \bmod q,\ \boldsymbol{\varpi},\ \boldsymbol{t}_{PID_s}\right)$.

Finally, the RSU id_1 sends the message $\boldsymbol{\varpi}$ to other RSUs in the group.

9) Trace and revocation: Once the RSU discovers malicious information from any vehicle, it signs the pseudonym with its private key and sends it to the KGC. After verifying the signature, the KGC performs the revocation operation. The RSU then submits the vehicle's pseudonym PID_p to the KGC. Next, KGC traces the real identity by calculating $ID_p = H_1(\boldsymbol{R}_T,\ PID_p^1) \oplus PID_p^2$. Then, the KGC updates the revocation parameters as follows:

(1) Sample a new PRSK $b' \xleftarrow{\$} S_1$.

(2) Compute $\boldsymbol{A}_K\boldsymbol{B}' = \overline{\boldsymbol{A}}' \bmod q$, where $\boldsymbol{B}' = \mathrm{rot}(b')$.

(3) Use the remaining RSKs to construct a new RP $f'(x) = \prod_{i=1}^{p-1}(x - r_{ID_i}) + b'$.

(4) Publish $f'(x)$ as the new RP and $\overline{\boldsymbol{A}}'$ as the new RPK.

VI. CORRECTNESS AND SECURITY PROOF

In this section, we provide a correctness analysis, formal security proof, and analysis using the Scyther tool, and informal security analysis of our scheme.

A. Correctness Analysis of Our LCMS

First, the signcryptext can be decrypted correctly according to the following derivation:

$$\begin{aligned}
&\hat{\boldsymbol{\tau}} = (\tau_1', \tau_2', \cdots, \tau_n')\\
&= \boldsymbol{c}_{\boldsymbol{id},3} - \boldsymbol{c}_1 - \boldsymbol{c}_2\\
&= \left\lfloor \frac{q'}{q}\overline{\boldsymbol{P}}_{\boldsymbol{id}}{}^T\boldsymbol{r} \right\rfloor + \left\lfloor \frac{q'}{2} \right\rfloor \boldsymbol{\tau} - \sum_{id_i \in \boldsymbol{id}} \boldsymbol{c}_{id_i,1} - \sum_{id_i \in \boldsymbol{id}} \boldsymbol{c}_{id_i,2}\\
&= \left\lfloor \frac{q'}{q}\overline{\boldsymbol{P}}_{\boldsymbol{id}}{}^T\boldsymbol{r} \right\rfloor + \left\lfloor \frac{q'}{2} \right\rfloor \boldsymbol{\tau} - \sum_{id_i \in \boldsymbol{id}} (\boldsymbol{D}_{id_i}{}^T\boldsymbol{c}_{\boldsymbol{id},1} + \boldsymbol{e}_{id_i,1})\\
&\qquad - \sum_{id_i \in \boldsymbol{id}} (\boldsymbol{C}_{id_i}{}^T\boldsymbol{c}_{\boldsymbol{id},2} + \boldsymbol{e}_{id_i,2})\\
&= \left\lfloor \frac{q'}{q}\overline{\boldsymbol{P}}_{\boldsymbol{id}}{}^T\boldsymbol{r} \right\rfloor + \left\lfloor \frac{q'}{2} \right\rfloor \boldsymbol{\tau} - \sum_{id_i \in \boldsymbol{id}} (\boldsymbol{D}_{id_i}{}^T\boldsymbol{c}_{\boldsymbol{id},1} + \boldsymbol{e}_{id_i,1})\\
&\qquad - \sum_{id_i \in \boldsymbol{id}} (\boldsymbol{C}_{id_i}{}^T\boldsymbol{c}_{\boldsymbol{id},2} + \boldsymbol{e}_{id_i,2})\\
&= \frac{q'}{q}\overline{\boldsymbol{P}}_{\boldsymbol{id}}{}^T\boldsymbol{r} - \boldsymbol{e}_{\overline{\boldsymbol{P}}_{\boldsymbol{id}}} + \left\lfloor \frac{q'}{2} \right\rfloor \boldsymbol{\tau}\\
&\qquad - \sum_{id_i \in \boldsymbol{id}} \left(\boldsymbol{D}_{id_i}{}^T \left\lfloor \frac{q'}{q} \times \boldsymbol{A}_K{}^T\boldsymbol{r} \right\rfloor + \boldsymbol{e}_{id_i,1} \right)\\
&\qquad - \sum_{id_i \in \boldsymbol{id}} \left(\boldsymbol{C}_{id_i}{}^T \lfloor \frac{q'}{q} \times \boldsymbol{A}_S{}^T\boldsymbol{r} \rceil + \boldsymbol{e}_{id_i,2} \right)\\
&= \frac{q'}{q}\overline{\boldsymbol{P}}_{\boldsymbol{id}}{}^T\boldsymbol{r} - \boldsymbol{e}_{\overline{\boldsymbol{P}}_{\boldsymbol{id}}} + \left\lfloor \frac{q'}{2} \right\rfloor \boldsymbol{\tau}\\
&\qquad - \sum_{id_i \in \boldsymbol{id}} \left(\boldsymbol{D}_{id_i}{}^T \left(\frac{q'}{q} \times \boldsymbol{A}_K{}^T\boldsymbol{r} - \boldsymbol{e}_{\boldsymbol{A}_K} \right) + \boldsymbol{e}_{id_i,1} \right)\\
&\qquad - \sum_{id_i \in \boldsymbol{id}} \left(\boldsymbol{C}_{id_i}{}^T \left(\frac{q'}{q} \times \boldsymbol{A}_S{}^T\boldsymbol{r} - \boldsymbol{e}_{\boldsymbol{A}_S} \right) + \boldsymbol{e}_{id_i,2} \right)\\
&= \frac{q'}{q}\overline{\boldsymbol{P}}_{\boldsymbol{id}}{}^T\boldsymbol{r} - \boldsymbol{e}_{\overline{\boldsymbol{P}}_{\boldsymbol{id}}} + \left\lfloor \frac{q'}{2} \right\rfloor \boldsymbol{\tau} - \frac{q'}{q} \sum_{id_i \in \boldsymbol{id}} (\boldsymbol{A}_K\boldsymbol{D}_{id_i})^T\boldsymbol{r}\\
&\qquad - \sum_{id_i \in \boldsymbol{id}} (-\boldsymbol{D}_{id_i}{}^T\boldsymbol{e}_{\boldsymbol{A}_K} + \boldsymbol{e}_{id_i,1})\\
&\qquad - \frac{q'}{q} \sum_{id_i \in \boldsymbol{id}} (\boldsymbol{A}_S\boldsymbol{C}_{id_i})^T\boldsymbol{r}\\
&\qquad - \sum_{id_i \in \boldsymbol{id}} (-\boldsymbol{C}_{id_i}{}^T\boldsymbol{e}_{\boldsymbol{A}_S} + \boldsymbol{e}_{id_i,2})\\
&= \frac{q'}{q}\overline{\boldsymbol{P}}_{\boldsymbol{id}}{}^T\boldsymbol{r} - \boldsymbol{e}_{\overline{\boldsymbol{P}}_{\boldsymbol{id}}} + \left\lfloor \frac{q'}{2} \right\rfloor \boldsymbol{\tau} - \frac{q'}{q}\overline{\boldsymbol{P}}_{\boldsymbol{id}}{}^T\boldsymbol{r}\\
&\qquad - \sum_{id_i \in \boldsymbol{id}} (-\boldsymbol{D}_{id_i}{}^T\boldsymbol{e}_{\boldsymbol{A}_K} + \boldsymbol{e}_{id_i,1})\\
&\qquad - \sum_{id_i \in \boldsymbol{id}} (-\boldsymbol{C}_{id_i}{}^T\boldsymbol{e}_{\boldsymbol{A}_S} + \boldsymbol{e}_{id_i,2})\\
&= \left\lfloor \frac{q'}{2} \right\rfloor \boldsymbol{\tau} - \boldsymbol{e}_{\overline{\boldsymbol{P}}_{\boldsymbol{id}}} - \sum_{id_i \in \boldsymbol{id}} (\boldsymbol{e}_{id_i,1} - \boldsymbol{D}_{id_i}{}^T\boldsymbol{e}_{\boldsymbol{A}_K})\\
&\qquad - \sum_{id_i \in \boldsymbol{id}} (\boldsymbol{e}_{id_i,2} - \boldsymbol{C}_{id_i}{}^T\boldsymbol{e}_{\boldsymbol{A}_S})\\
&= \left\lfloor \frac{q'}{2} \right\rfloor \boldsymbol{\tau} + \triangle,
\end{aligned}$$

where $\triangle = -\boldsymbol{e}_{\overline{\boldsymbol{P}}_{\boldsymbol{id}}} - \sum_{id_i \in \boldsymbol{id}} (\boldsymbol{e}_{id_i,1} - \boldsymbol{D}_{id_i}{}^T\boldsymbol{e}_{\boldsymbol{A}_K}) - \sum_{id_i \in \boldsymbol{id}} (\boldsymbol{e}_{id_i,2} - \boldsymbol{C}_{id_i}{}^T\boldsymbol{e}_{\boldsymbol{A}_S})$, $\boldsymbol{e}_{\overline{\boldsymbol{P}}_{\boldsymbol{id}}} = \frac{q'}{q}\overline{\boldsymbol{P}}_{\boldsymbol{id}}{}^T\boldsymbol{r} - \left\lfloor \frac{q'}{q}\overline{\boldsymbol{P}}_{\boldsymbol{id}}{}^T\boldsymbol{r} \right\rfloor \in [-\frac{1}{2}, \frac{1}{2})^n$, $\boldsymbol{e}_{\boldsymbol{A}_K} = \frac{q'}{q}\boldsymbol{A}_K{}^T\boldsymbol{r} - \left\lfloor \frac{q'}{q}\boldsymbol{A}_K{}^T\boldsymbol{r} \right\rfloor \in [-\frac{1}{2}, \frac{1}{2})^m$, and $\boldsymbol{e}_{\boldsymbol{A}_S} = \frac{q'}{q}\boldsymbol{A}_S{}^T\boldsymbol{r} - \left\lfloor \frac{q'}{q}\boldsymbol{A}_S{}^T\boldsymbol{r} \right\rfloor \in [-\frac{1}{2}, \frac{1}{2})^m$. So, we can compute $\|\triangle\|_\infty \le \frac{1}{2} + |\boldsymbol{id}| \times (\frac{3}{2}\sigma m + 4\sigma_{flood})$.

According to the above formula, $\boldsymbol{\tau}$ can be correctly decrypted when $\|\triangle\|_\infty < \left\lfloor \frac{q'}{4} \right\rfloor$. If $\tau_j' \in [-\left\lfloor \frac{q'}{4} \right\rfloor, \left\lfloor \frac{q'}{4} \right\rfloor)$, $\tau_j = 0$, otherwise $\tau_j = 1$, Thereby, $\boldsymbol{\tau}$ can be restored, $\boldsymbol{\varpi}$ can be restored by $\boldsymbol{\varpi} = \boldsymbol{x} \oplus H_5(\boldsymbol{\tau})$.

Second, $f\left(r_{ID_j}\right) = \prod_{i=1}^{p}\left(r_{ID_j} - r_{ID_i}\right) + b = b$, $j \in [p]$, $\boldsymbol{B} = \mathrm{rot}(b)$.

Third, if $\boldsymbol{c}_{PID_s} \neq H_2\left(\begin{bmatrix}\boldsymbol{A}_K\boldsymbol{y}_{PID_s,1}\\ \boldsymbol{A}_S\boldsymbol{y}_{PID_s,2}\end{bmatrix} \bmod q, \boldsymbol{\varpi}, \boldsymbol{t}_{PID_s}\right)$ or $\left\|\boldsymbol{z}_{PID_s,1}\right\| > 2\sigma_s\sqrt{m}$ or $\left\|\boldsymbol{z}_{PID_s,1}\right\| = 0$ or $\left\|\boldsymbol{z}_{PID_s,2}\right\| > 2\sigma_s\sqrt{m}$ or $\left\|\boldsymbol{z}_{PID_s,2}\right\| = 0$, the signcryptext is illegal. Otherwise, combining pseudonyms PID_s, the public key $\boldsymbol{P}_{PID_s}$, the message $\boldsymbol{\varpi}$, and the current timestamp $\boldsymbol{t}_{PID_s}$, the legality of our scheme mainly relies on the equation $\boldsymbol{c}_{PID_s} = H_2\left(\begin{bmatrix}\boldsymbol{A}_K\boldsymbol{y}_{PID_s,1}\\ \boldsymbol{A}_S\boldsymbol{y}_{PID_s,2}\end{bmatrix} \bmod q, \boldsymbol{\varpi}, \boldsymbol{t}_{PID_s}\right)$. The detailed steps are described as follows:

(1) Due to

$$\boldsymbol{A}_K\boldsymbol{z}_{PID_s,1} - (H_3(PID_s) + \bar{\boldsymbol{A}})\boldsymbol{c}_{PID_s} \bmod q$$
$$= \boldsymbol{A}_K\left(\left(\boldsymbol{D}_{PID_s} + \boldsymbol{B}\right)\boldsymbol{c}_{PID_s} + \boldsymbol{y}_{PID_s,1}\right) - (H_3(PID_s) + \bar{\boldsymbol{A}})\boldsymbol{c}_{PID_s} \bmod q$$
$$= \boldsymbol{A}_K\left(\boldsymbol{D}_{PID_s} + \boldsymbol{B}\right)\boldsymbol{c}_{PID_s} + \boldsymbol{A}_K\boldsymbol{y}_{PID_s,1} - (H_3(PID_s) + \bar{\boldsymbol{A}})\boldsymbol{c}_{PID_s} \bmod q$$
$$= \left(\boldsymbol{A}_K\boldsymbol{D}_{PID_s} + \boldsymbol{A}_K\boldsymbol{B}\right)\boldsymbol{c}_{PID_s} + \boldsymbol{A}_K\boldsymbol{y}_{PID_s,1} - (H_3(PID_s) + \bar{\boldsymbol{A}})\boldsymbol{c}_{PID_s} \bmod q$$
$$= (H_3(PID_s) + \bar{\boldsymbol{A}})\boldsymbol{c}_{PID_s} + \boldsymbol{A}_K\boldsymbol{y}_{PID_s,1} - (H_3(PID_s) + \bar{\boldsymbol{A}})\boldsymbol{c}_{PID_s} \bmod q$$
$$= \boldsymbol{A}_K\boldsymbol{y}_{PID_s,1} \bmod q, \text{ and}$$
$$\boldsymbol{A}_S\boldsymbol{z}_{PID_s,2} - \boldsymbol{P}_{PID_s}\boldsymbol{c}_{PID_s} \bmod q$$
$$= \boldsymbol{A}_S\left(\boldsymbol{D}_{PID_s}\boldsymbol{c}_{PID_s} + \boldsymbol{y}_{PID_s,2}\right) - \boldsymbol{P}_{PID_s}\boldsymbol{c}_{PID_s} \bmod q$$
$$= \boldsymbol{A}_S\boldsymbol{D}_{PID_s}\boldsymbol{c}_{PID_s} + \boldsymbol{A}_S\boldsymbol{y}_{PID_s,2} - \boldsymbol{P}_{PID_s}\boldsymbol{c}_{PID_s} \bmod q$$
$$= \boldsymbol{P}_{PID_s}\boldsymbol{c}_{PID_s} + \boldsymbol{A}_S\boldsymbol{y}_{PID_s,2} - \boldsymbol{P}_{PID_s}\boldsymbol{c}_{PID_s} \bmod q$$
$$= \boldsymbol{A}_S\boldsymbol{y}_{PID_s,2} \bmod q,$$
$$\boldsymbol{c}_{PID_s} = H_2\left(\begin{bmatrix}\boldsymbol{A}_K\boldsymbol{z}_{PID_s,1} - (H_3(PID_s) + \bar{\boldsymbol{A}})\boldsymbol{c}_{PID_s} \\ \boldsymbol{A}_S\boldsymbol{z}_{PID_s,2} - \boldsymbol{P}_{PID_s}\boldsymbol{c}_{PID_s}\end{bmatrix} \bmod q, \boldsymbol{\varpi}, \boldsymbol{t}_{PID_s}\right).$$

(2) $\boldsymbol{z}_{PID_s,1} = (\boldsymbol{D}_{PID_s} + \boldsymbol{B})\boldsymbol{c}_{PID_s} + \boldsymbol{y}_{PID_s,1}$, where $\boldsymbol{y}_{PID_s,1}$ obeys the $\mathcal{D}_{\mathbb{Z}^m,\sigma_s}$ distribution. According to the rejection sampling algorithm [28], it can be known that the distribution satisfied by the output $\boldsymbol{z}_{PID_s,1} = (\boldsymbol{D}_{PID_s} + \boldsymbol{B})\boldsymbol{c}_{PID_s} + \boldsymbol{y}_{PID_s,1}$ with the probability of $\min(1, \frac{\mathcal{D}_{\mathbb{Z}^m,\sigma_s}(\boldsymbol{z}_{PID_s,1})}{M_1\mathcal{D}_{\mathbb{Z}^m,\sigma_s,(\boldsymbol{D}_{PID_s}+\boldsymbol{B})\boldsymbol{c}_{PID_s}}(\boldsymbol{z}_{PID_s,1})})$ is statistically close to the $\mathcal{D}_{\mathbb{Z}^m,\sigma_s}$ distribution (statistical distance is $2^{-100}/M_1$, where $M_1 \approx 3$), so according to Lemma 1, $\boldsymbol{z}_{PID_s,1}$ satisfies $\|\boldsymbol{z}_{PID_s,1}\| \le 2\sigma_s\sqrt{m}$ with a probability of $1 - 2^{-m}$. $\boldsymbol{z}_{PID_s,2}$ can be analyzed using the same method, which will not be repeated here.

In conclusion, our signcryptext verification is correct.

B. Formal Security Proof

The security of the LCMS needs to be proven against two types of adversaries. The type I adversary involves an "external" adversary, where the adversary can replace vehicles' public key, but the PPK of any vehicle cannot be obtained. The type II adversary is an "internal" adversary, where the adversary can only obtain the MSK of KGC, but cannot replace the public key of any vehicle. The security of our scheme is captured in the following theorems. We leave the proofs of these theorems in Appendix.

Game I, type I adversary:

Theorem 1. It is assumed that the hash function can be regarded as a random oracle, and the $SIS_{q,n,2m,\beta}$ is difficult (where $\beta = \sqrt{(4\sigma_s\sqrt{m} + 2\sigma\kappa\sqrt{m} + 2\kappa\sqrt{m})^2 + (4\sigma_s\sqrt{m} + 4\sigma\kappa\sqrt{m})^2}$). If the signcryptext of any user ID on any message can be successfully forged by the adversary $\mathcal{A}_1$ in probabilistic polynomial time (PPT), then there is an algorithm to solve the $SIS_{q,n,2m,\beta}$ problem with a non-negligible probability. In other words, under the assumption that the $SIS_{q,n,2m,\beta}$ problem is hard, our scheme satisfies EUF-aID-CMA under the random oracle model.

Game I, type II adversary:

Theorem 2. Assuming that the hash function can be regarded as a random oracle, and the small integer solution problem $SIS_{q,n,2m,\beta}$ is difficult (where $\beta = \sqrt{(4\sigma_s\sqrt{m} + 2\sigma\kappa\sqrt{m} + 2\kappa\sqrt{m})^2 + (4\sigma_s\sqrt{m} + 4\sigma\kappa\sqrt{m})^2}$), if the signcryptext of any user ID on any message in our scheme can be successfully forged by the adversary $\mathcal{A}_2$ in PPT, there is an algorithm that can solve the $SIS_{q,n,2m,\beta}$ problem with a non-negligible probability. In other words, under the assumption that the $SIS_{q,n,2m,\beta}$ problem is hard, our scheme satisfies EUF-aID-CMA under the random oracle model.

Game II, Type I adversary:

Theorem 3. It is assumed that the hash function can be regarded as a random oracle, and the Decision $SIS_{q,n,m,\beta}$ problem is difficult. If there is an adversary $\mathcal{A}_1$ that can overcome the IND-aID-CCA2 security of our scheme, then the challenger $\mathcal{C}$ can solve the Decision $SIS_{q,n,m,\beta}$ problem in PPT. That is, in the ROM, our scheme satisfies IND-aID-CCA2.

Game II, Type II adversary:

Theorem 4. It is assumed that the hash function can be regarded as a random oracle, and the Decision $SIS_{q,n,m,\beta}$ problem is difficult. If there is an adversary $\mathcal{A}_2$ that can overcome the IND-aID-CCA2 security of our scheme, then the challenger $\mathcal{C}$ can solve the Decision $SIS_{q,n,m,\beta}$ problem in PPT. That is, in the ROM, our scheme satisfies IND-aID-CCA2.

Proofs: The formal proofs of these theorems are provided in Appendix.

C. Formal Security Analysis Using the Scyther

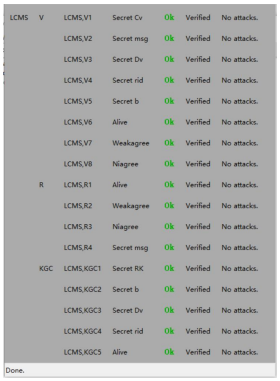


Fig. 6. Analysis result using the Scyther tool.

We conducted a formal analysis of our LCMS using the latest stable version of Scyther (v1.1.3) [32], released April 4, 2014. Scyther is an automated security protocol verification tool. To validate the security of our scheme, we write it in the security protocol description language. We define the roles as follows: V for the vehicle, R for the RSU, and KGC for the traffic control center. Fig. 6 depicts the Scyther-generated plot of the signcryption mechanism in action among a vehicle (V), an RSU, and a KGC. This plot shows the successful security verification of our scheme.

D. Informal Security Analysis

In this subsection, the informal security analysis shows our scheme follows the expected security goals and is resistant to several potential attacks. Let the following abbreviations

represent key security properties: Aut-Int for authentication and integrity, Con for confidentiality, Ano for vehicle anonymity, WeUnl for weak unlinkability, Tra for traceability, and Rev for revocability. Imm-Imp, Imm-Mod, Imm-Man, Imm-Rep, Imm-KT, and Imm-KE denote immunity to impersonation, modification, man-in-the-middle, replay, malicious KGC attacks, and key escrow, respectively. Table III elaborates the comparison of these properties between our scheme and similar schemes [15]-[22], [30], and [31]. The comparison shows that our scheme exhibits more comprehensive security, making it more suitable for IoV.

1) Authentication and integrity: According to our security proof, our scheme achieves EUF-aID-CMA and IND-aID-CCA2 against both adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$. Therefore, the signcryptext from a registered, non-revoked vehicle can pass the unsigncryption operation of RSUs. In addition, if the signcryptext is tampered with during transmission, the unsigncryption operation will fail. Thus, our scheme satisfies data authentication and integrity.

2) Confidentiality: According to our security proof, our scheme achieves EUF-aID-CMA and IND-aID-CCA2 against both adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$. Therefore, only the RSUs specified by the vehicle can obtain the plaintext through the unsigncryption operation. Thus, our scheme satisfies data confidentiality.

3) Vehicle anonymity: Our scheme uses a pseudonym mechanism ($PID = (PID^1,\ PID^2 = H_1(\boldsymbol{R}_K,\ PID^1) \oplus ID)$). This ensures RSUs can only obtain the pseudonym of the vehicle, not its real identity. If an adversary wants to obtain ID through PID, he/she must obtain the MSK $\boldsymbol{R}_K$. Against $\mathcal{A}_1$, this is impossible. Against $\mathcal{A}_2$, while it theoretically can access MSK $\boldsymbol{R}_K$, the trusted-KGC assumption in our model prevents actual MSK leakage. Therefore, under the standard trusted-KGC assumption, neither adversary can break vehicle anonymity. Thus, our scheme satisfies this property.

4) Weak unlinkability: First, our scheme employs a pseudonym mechanism, preventing RSUs from linking vehicular messages to the real identity of vehicles. Second, in our scheme, each vehicle can apply for multiple pseudonyms, with each pseudonym corresponding to distinct public keys, private keys, and PPKs. Vehicles can alternate among different pseudonyms to avoid consecutive messages being correlated by malicious adversaries. Combining these two aspects, our scheme satisfies weak unlinkability.

5) Traceability: Although our scheme satisfies vehicle anonymity, when a malicious vehicle is found, KGC can recover the true identity of the vehicle by calculating $ID = H_1(\boldsymbol{R}_K,\ PID^1) \oplus PID^2$. Therefore, our scheme satisfies traceability.

6) Revocability: After tracing back the identity of the malicious vehicle, KGC can update the PRSK and the RP to revoke the signcryption rights of the malicious vehicle. The specific operation is as follows:

The KGC samples a new PRSK $b' \in \mathbb{R}_q$, computes $\boldsymbol{A}_K \boldsymbol{B}' = \overline{\boldsymbol{A}}'$ ($\boldsymbol{B}' = \mathrm{rot}(b')$), and uses the remaining RSKs to construct a new RP $f'(x) = \prod_{i=1}^{p-1}(x - r_{ID_i}) + b'$. Finally, KGC publishes $f'(x)$ as the new RP and $\overline{\boldsymbol{A}}'$ as the new RPK. At this point, the revoked vehicle can no longer obtain the new PRSK and cannot generate new legal signcryptext, so our scheme meets revocability.

7) Avoid key escrow: Because our scheme is constructed based on a certificateless structure, our scheme can avoid key escrow.

8) Resilience to specific attacks:

(1) Resilience to impersonation attacks: Because the PPK, the SV, and the RSK of vehicles cannot be obtained by any adversary, our scheme is against the impersonation attacks.

(2) Resistance to modification and man-in-the-middle attacks: Since our scheme has achieved authentication and integrity. As a result, our scheme is against modification and man-in-the-middle attacks.

(3) Resistance to replay attacks: In our scheme, the current timestamp needs to be added, and RSU checks the freshness of the timestamp. Therefore, our scheme is against the replay attack.

(4) Resilience to malicious KGC attacks: Because our scheme is constructed based on the certificateless structure, our scheme is resistant to malicious KGC in the key generation process.

VII. PERFORMANCE EVALUATION

In this section, we analyze the performance of the computation overhead and the storage overhead, the communication burden, and the RSU serving capability of our LCMS and compare the computation overhead and the storage overhead with that of related works [15]-[22], [30], and [31].

TABLE III COMPARISON BETWEEN OUR SCHEME AND THE OTHER SCHEMES IN TERMS OF SECURITY ATTRIBUTES AND FUNCTIONS

Schemes	Ullah et al. [15]	Wang et al. [16]	Liang et al. [17]	Liang et al. [18]	Su et al. [19]	Le et al. [20]	Yu et al. [21]	Yu et al. [22]	Xu et al. [30]	Bai et al. [31]	Our scheme
Multi-Receiver	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No	Yes
Secure in the post-quantum era	No	No	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Distributed decryption	No	No	No	No	No	No	No	No	No	No	Yes
Aut-Int	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Con	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Ano	No	No	Yes	No	Yes	No	No	No	No	No	Yes
WeUnl	No	No	Yes	No	No	No	No	No	No	No	Yes
Tra	No	No	Yes	No	Yes	No	No	No	No	No	Yes
Rev	No	No	Yes	No	Yes	No	No	No	No	No	Yes
Imm-KE	No	No	Yes	No	Yes	No	Yes	No	Yes	Yes	Yes

Imm-Imp	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Imm-Mod	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Imm-Man	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Imm-Rep	No	No	Yes	Yes	Yes	No	No	No	No	No	Yes
Imm-MK	No	No	Yes	No	Yes	No	Yes	No	Yes	Yes	Yes

A. Computation Overhead and Storage Overhead

Based on expensive mathematical operations, we compare the costs (storage, time) of our scheme against the existing schemes [17], [20], [21], [22], [30], and [31]. Table VI compares the time cost of key generation, signcryption, and unsigncryption of our scheme against the existing schemes. Table VII compares the storage cost of the public key, the private key, and the signcryptext of our scheme against the existing schemes. Compared to the classical number-theoretic scheme [17], our scheme incurs larger storage overhead and appears slower. However, scheme [17] cannot withstand quantum computer attacks. Compared to the scheme based on the structured lattice [31], our scheme is safer but slower. This is because structured lattice-based schemes can be accelerated using the number-theoretic transform (NTT) algorithm [33]. However, the scheme based on the structured lattice seems less secure than the scheme based on the random lattice (structured lattices are more susceptible to being broken by quantum algorithms than random lattices [50]).

We implement our scheme using the SageMath computer algebra system, version 10.7 [37]. Experiments were conducted on a Lenovo ThinkPad T480s (Intel i7-8650U, 16GB RAM) running Windows 10. Table IV describes common parameters; selection methods follow Refs. [25], [26], [28], [35], [40], and [41]. We use SHAKE-256 [38] to instantiate the hash operation and the Lattice Estimator tool [39] for parameter's security evaluation. The result of the lattice estimator on SIS (n =800, $q = 2^{69} + 978689$, $2 \times m = 4n\lceil \log q \rceil$ =220800, $\text{length} - \text{bound} = \sqrt{(4\sigma_s\sqrt{m} + 2\sigma\kappa\sqrt{m} + 2\kappa\sqrt{m})^2 + (4\sigma_s\sqrt{m} + 4\sigma\kappa\sqrt{m})^2}$ =185 925941874, $\text{norm} = 2$) is as follows:

lattice:: rop: ≈2^129.6, red: ≈2^129.6, δ: 1.004391, β: 347, d: 2952, tag: euclidean

Result analysis:

Lattice reduction attack: The computational cost is approximately $2^{129.6}$ operations. The computational cost is extremely high, far exceeding the practical computational capabilities available now and for the foreseeable future.

Table V reports the time overhead of algorithm primitives and our unique algorithms. Figs. 7-8 compare our scheme with schemes [20], [21], [22], [30], and [31], showing small time and storage overhead at large parameters. In Fig. 7 and Fig. 8, measurements are log-transformed in base 2 to accommodate the extreme range in values. Fig. 7 compares the execution times of parameter generation, signcryption, and unsigncryption. Fig. 8 compares public key, private key, and signcryptext sizes. After comparison, under large parameters, our scheme achieves the smallest private key size, signcryptext size, signcryption time, and unsigncryption time among all schemes, except scheme [31]. The key generation time and the public key size are nearly the smallest and similar to scheme [20]. Besides, our scheme is also more secure and feature-rich.

Fig. 9 shows the time overhead for signcryption, unsigncryption, join, and revocation as the number of vehicles and RSUs increases. When $p = 300$, $l = 300$, these overheads remain acceptable, not exceeding 56 seconds.

TABLE IV COMMON PARAMETERS SETTING

Parameter	Definition	Sample
λ	The security parameter	128
n	The matrix dimension	800
κ	The challenge bound, $\binom{n}{\kappa} 2^{\kappa} \geq 2^{128}$	17
q	The prime modulus q, $q \equiv 1 \pmod{2m}$	$2^{69} + 978689$
$q^{'}$	The LWR modulus, $\frac{q}{q^{'}} \geq \sqrt{n}$	$2^{41} + 1$
m	The matrix dimension, $m = 2n\lceil \log q \rceil$	110400
$\bar{m}$	The matrix dimension, $\bar{m} = n\lceil \log q \rceil$	55200
w	The matrix dimension, $w = n\lceil \log q \rceil$	55200
N	The maximum degree of the polynomial	800
σ	The SIS Gaussian parameter, $\sigma \geq \sqrt{n \log q} \times \omega(\sqrt{\log n})$	705
σ_s	The signature Gaussian parameter, $\sigma_s = \alpha \|\|\boldsymbol{Cc}\|\| = 24\sigma\kappa\sqrt{m}$	95551056
σ_{lwe}	The LWE Gaussian parameter, $\sigma_{lwe} = \frac{q}{2q^{'}\sqrt{3}}$	77490641
σ_{flood}	The flooding noise Gaussian parameter, $\sigma_{flood} = \sqrt{2}\sigma_{lwe}$	109588316
l	The number of the receivers, $l = \left\lfloor \frac{q^{'}}{4} \right\rfloor / (\frac{3}{2}\sigma m + 4\sigma_{flood}) \approx 990$ under the current parameter	l
p	The number of registered vehicles (we assume that each registered vehicle applies for three pseudonyms)	p ($p \times 3$)
M	The rejection sampling times, $M = \mathrm{e}^{\frac{12}{\alpha} + \frac{1}{2\alpha^2}}$, α=12	≈2.7

TABLE V SOME ALGORITHMIC TIME OVERHEAD

Algorithm	GenTrap	SampleMat	SampleGaussian/Invert	Join	Trace and revocation
Time overhead	$n\bar{m}w$	n^2m	nm	$nm + (p+2) \times (p-1) \times m\log m$	$nm + pm\log m$

Notes: During the Join phase and the Trace and revocation phase, we use the NTT algorithm [33] to optimize polynomial multiplication. During the Trace and revocation phase, we use polynomial division to remove the revoked RSK. The Invert algorithm is used in Ref. [20].

TABLE VI TIME COST COMPARISON BETWEEN OUR SCHEME AND OTHER SCHEMES

Schemes	Time cost for the generation of the public/private key	Time cost for the signcryption	Time cost for the unsigncryption
Our scheme (random lattice)	$n\bar{m}w + 3n^2m$	$ln^2 + 2nm + n^2 + M(4nm + (\log p + p)m\log m)$	$2nm + 4n^2 + 2lnm + 2ln$
Liang et al. (based on ECC) [17]	$5t_m + t_a + 2t_h$	$9t_m + 4t_a + 5t_h$	$6t_m + 5t_a + 5t_h$

Le et al. [20]	$n\bar{m}w$	$3n^2\bar{m} + 2n\bar{m}w + 3nm + 3n\bar{m} + 2\bar{m}$	$2n^2\bar{m} + n\bar{m}w + nm + nm\bar{m} + m\bar{m} + 2\bar{m}$
Yu et al. [21]	$3nm^2 + m^3$	$nm^2 + nm + 2m + n$	$2m^2 + n + m$
Yu et al. [22]	nm^2	$3nm + nm^2 + m^2 + n$	$2m^3 + m^2 + nm^2 + n + nm$
Xu et al. [30]	$nm^2 + 2nm$	$nM(12nm + 3m + n)$	$n(n + m + 12nm)$
Bai et al. (structured lattice) [31]	$(2n^2m + M(2n^2 + 3nm + 2n^2m) + n^2 + 2nm)N\log N$	$(4n^2 + 4nm + n + m + M(4n + 2m + nm))N\log N$	$(6n + 3m + 4n^2 + 4nm)N\log N$

Notes: t_m represents the time cost for scale multiplication on ECC, t_a represents the time cost for point addition on ECC, t_h represents the time cost for the hash operation. $m = 2n\lceil\log q\rceil$ for our scheme, and schemes [20] and [31], $m = 6n\lceil\log q\rceil$ for scheme [21], $m = 5n\lceil\log q\rceil$ for schemes [22] and [30], $n = 1$ for scheme [31]. The scheme [30] only signcrypts 1 bit each time, while other schemes signcrypt n bits each time. To maintain fairness, we multiply the signcryption and unsigncryption time overhead of scheme [30] by n. To compute $f(r_{ID})$, we can use the method described in Refs. [33] [34], which has a time complexity of $O((\log p + p)m\log m)$, where $O(m\log m)$ is the time complexity of multiplying two degree-m polynomials using the NTT [33], $O(\log p)$ is the time complexity of an element raised to the power of p using fast exponentiation [34].

TABLE VII STORAGE COST COMPARISON BETWEEN OUR SCHEME AND OTHER SCHEMES

Schemes	Storage cost for the public key	Storage cost for the private key	Storage cost for the signcryptext
Our scheme (random lattice)	$(2nm + n^2)\lceil\log q\rceil$	$\bar{m}w\lceil\log(4\sigma)\rceil + 2nm\lceil\log(4\sigma)\rceil$	$(2m + n)\log q' + 2m\lceil\log(4\sigma_s)\rceil + n$
Liang et al. (based on ECC) [17]	$2\lvert G\rvert$	$2\lceil\log q\rceil$	$3\lvert G\rvert + 3\lceil\log q\rceil$
Le et al. [20]	$2nm\lceil\log q\rceil$	$2\bar{m}w\lceil\log(4\sigma)\rceil$	$2m\lceil\log q\rceil + 2\bar{m}\lceil\log q\rceil + (4m + \bar{m})\lceil\log(4\sigma)\rceil$
Yu et al. [21]	$2nm\lceil\log q\rceil$	$m^2\lceil\log(4\sigma)\rceil + m^2\lceil\log q\rceil$	$n\lceil\log q\rceil + 2m\lceil\log q\rceil + m\lceil\log(4\sigma)\rceil$
Yu et al. [22]	$nm\lceil\log q\rceil$	$m^2\lceil\log(4\sigma)\rceil$	$m\lceil\log q\rceil$
Xu et al. [30]	$(nm + m(1 + n))\lceil\log q\rceil$	$(m^2 + m + n)\lceil\log(4\sigma)\rceil$	$n(n + 2m + 1)\lceil\log q\rceil$
Bai et al. (structured lattice) [31]	$(2nm + m + n)N\lceil\log q\rceil$	$(2nm + 3n + m)N\lceil\log(4\sigma)\rceil$	$(3n + m + 1)N\lceil\log q\rceil + (3n + m)N\lceil\log(4\sigma_s)\rceil$

Notes: $\lvert G\rvert$ represents the size of an additive elliptic curve group G with prime order q in ECC-based schemes. $m = 2n\lceil\log q\rceil$ for our scheme, and schemes [20] and [31], $m \approx 6n\lceil\log q\rceil$ for scheme [21], $m \approx 5n\lceil\log q\rceil$ for schemes [22] and [30], $n = 1$ for scheme [31]. The scheme [30] only signcrypts 1 bit each time, while other schemes signcrypt n bits each time. To maintain fairness, we multiply the signcryptext size of scheme [30] by n.

TABLE VIII COMMUNICATION OVERHEAD OF OUR SCHEME

The different part	C1	C2	C3	C4
Communication overhead	$(l + 3p) \times nm\lceil\log(4\sigma)\rceil$	$p \times m\lceil\log q\rceil$	$(2m + n)\lceil\log q'\rceil + 2m\lceil\log(4\sigma_s)\rceil + n$	$(l - 1) \times (2n\lceil\log q'\rceil + 2m\lceil\log q'\rceil)$

Notes: Let C1–C4 denote the communication costs (in bits) for the following four parts, respectively: C1: between the KGC and the vehicle/RSU during key generation; C2: while the KGC distributes the RSKs; C3: when the vehicle transmits the signcryptext; C4: during the distributed decryption process.

B. Communication Overhead

In our scheme, the total communication overhead consists of four components (C1–C4):

C1: Between KGC and the vehicle/RSU during key-pair generation;

C2: When KGC distributes the RSK;

C3: When the vehicle transmits the signcryptext;

C4: During the process of distributed decryption.

Table VIII provides formulas for these partial communication overheads. Fig. 10 shows the communication overhead under different numbers of vehicles and RSUs. The results show that our scheme incurs small communication overhead at large parameters. Note that C1 is large but only needs to be calculated once. Thus, our scheme is also practical.

Communication overhead between RSUs and the vehicles is constant. However, the overhead between RSUs increases linearly with the number of RSUs.

For RSU-to-RSU communication, RSUs are mostly connected via wired optical fiber or 5G private networks. These have large bandwidth (up to Gbps), low latency (<10 ms), and strong stability. Therefore, the communication overhead defect between our RSUs can be compensated for by hardware.

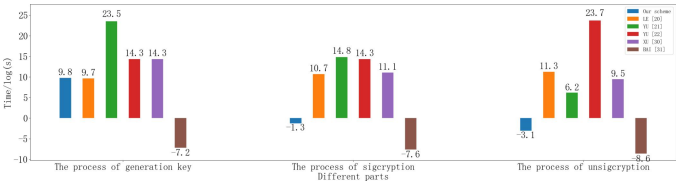


Fig. 7. Time cost comparison (binary log-transformed) for the generation of the public/private key, the signcryption, and the unsigncryption for our scheme and schemes [20], [21], [22], [30], and [31] (for the sake of fairness, we set $p = 1$, $l = 1$ here).

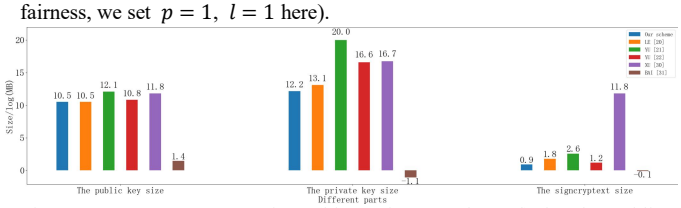


Fig. 8. Storage cost comparison (binary log-transformed) for the public key, the private key, and the signcryptext for our scheme and schemes [20], [21], [22], [30], and [31].

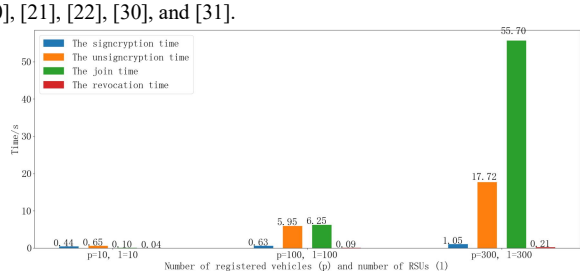


Fig. 9. Time cost for the signcryption process, the unsigncryption process, the join process, and the revocation process under different numbers of vehicles and RSUs.

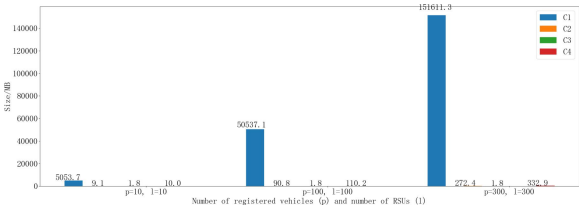


Fig.10. Communication overhead under different numbers of vehicles and RSUs.

Let C1–C4 denote the communication costs (in MB) for the following four parts, respectively: C1: between the KGC and the signer during key generation; C2: while the KGC distributes the RSK; C3: when the vehicle transmits the signcryptext; C4: during the distributed decryption process.

C. RSU Serving Capability

The RSU's serving capability [17] is defined as: $SC_{RSU} = \frac{pr \times r}{v \times d \times T_{total}}$. In this formula, pr is the probability of successful unsigncryption, r is the RSU coverage range, v is vehicle speed (40–60 km/h), and d is vehicle density (100–200/km^2). T_{total} includes the transmission time T_{tr} and the unsigncryption time T_{un}, measured under p=10, l=10.

According to Fig. 9 and Fig. 10 under p=10, l=10, we obtain T_{un}=0.65 s. The signcryptext communication overhead is 1.8 MB. Assuming the vehicle communication channel (IEEE 802.11p) bandwidth is 27 Mbps, then T_{tr} =(1.8 × 8 Mbits)/(27 Mbps) ≈ 0.53 s. Therefore, $T_{total} = T_{tr} + T_{un}$=1.18 s.

Fig. 11 and Fig. 12 present SC_{RSU} results for r=500 m and r=1000 m, respectively, showing how it varies with v and d. The figures demonstrate that SC_{RSU} is inversely proportional to both v and d. Therefore, by decreasing vehicle density without reducing the speed of vehicles, the RSU's serving capability can be increased in our LCMS.

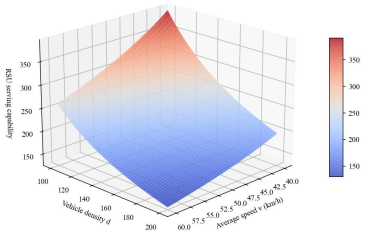


Fig. 11. The RSU serving capability with r=500.

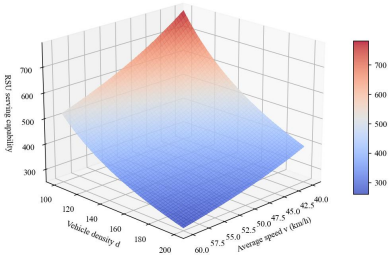


Fig. 12. RSU serving capability with r=1000.

VIII. CONCLUSION AND FUTURE WORK

In summary, we propose an LCMS for IoV scenarios. Our scheme integrates numerous properties: post-quantum security, vehicle anonymity, weak unlinkability, revocability, distributed decryption, and certificatelessness. These features comprehensively mitigate malicious behavior by all IoV components (KGC, RSUs, and vehicles). We leverage the LWR problem to reduce expensive discrete Gaussian sampling, along with ciphertext packing techniques, to reduce time overhead, signcryptext size and communication overhead, and use MP12 trapdoor to reduce the overall storage overhead and time overhead of our scheme. Both informal and formal security proof show our scheme satisfies EUF-aID-CMA and IND-aID-CCA2 in the ROM, and robust against IoV-specific attacks. Experiments show that our scheme exhibits more comprehensive security features and better time and storage efficiency than recent random lattice-based signcryption schemes. At large parameters, our scheme incurs only a small communication overhead, proving its practicality for IoV.

Future work may investigate replacing the random lattice with the structured lattice. The structured lattice offers lower security than the random lattice [50], but consumes less storage and time overhead. Currently, the KGC must recompute the RP after every update. This costs time. Moreover, the KGC must store each legitimate signer's RSK. As the user base grows, so does storage demand. Future work will optimize RP updates to save time. We will also seek smaller RSKs and lighter management to reduce space.

Tamper-Proof Device (TPD) storage feasibility discussion: based on the systematic optimization approach, the TPD storage requirements can be practically addressed through a hybrid TPD-Trusted Execution Environment (TEE) architecture that accommodates the simultaneous use of all signature components. For each pseudonym, the two $m \times n$ secret matrices $\boldsymbol{C}_{PID}$ and $\boldsymbol{D}_{PID}$ total ~160 MB, while the vehicle-unique RSK r_{ID} (a degree-m polynomial requiring ~0.9 MB) is stored separately as it is not part of the pseudonym secret key. The TPD stores only a 256-bit Advanced Encryption Standard master key (32 bytes) in its secure Non-Volatile Memory (608 KB–2 MB), with both

encrypted pseudonym secret matrices and the encrypted r_{ID} residing in external Embedded Multi-Media Card/Solid State Drive storage. During signature generation, the TPD derives the decryption key and coordinates with the TEE to establish a secure memory region protected by memory encryption (Total Memory Encryption). The TPD decrypts the complete $\boldsymbol{C}_{PID}$ and $\boldsymbol{D}_{PID}$, and vehicle-specific r_{ID} into the TEE-secured Random Access Memory (RAM) (tens of GB, transparently encrypted by hardware memory encryption), where the signature algorithm executes with all three components simultaneously present in plaintext. The memory encryption key, managed exclusively by the TPD, ensures that even physical extraction of Dynamic RAM chips yields only ciphertext. Upon signature completion, the TPD commands immediate hardware-zeroization of the secure memory region, leaving no residual plaintext. This architecture achieves Electronic- safety Vehicle Intrusion Protected Applications Full Hardware Security Module-equivalent protection for the root key material while practically accommodating the large key sizes inherent to lattice-based pseudonym systems.

Regarding computational overhead, the one-time key generation cost is substantial ($\approx$891 seconds per pseudonym, Fig. 7). This process is performed offline, and each vehicle only needs to apply for a few pseudonyms (3-5) to ensure weak unlinkability, so this will not affect the real-time performance of the V2X system.

Regarding join and revocation operations, under the condition of 300 vehicles and 300 RSUs, Fig. 9 indicates that only 0.2 seconds are required. The join operation depicted in the figure is performed offline; that is, all 300 vehicles join simultaneously during the initialization phase. In a real-world scenario, however, the joining or revocation of a single vehicle takes merely 0.2 seconds, a performance level that effectively meets the real-time requirements of vehicular network environments.

ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China (Grant No. 62572025), Beijing Natural Science Foundation (Grant No. 4242024), Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing (Grant No. GJJ-25-xxx), the Fundamental Research Funds for the Central Universities, China, Wenzhou Major Scientific and Technological Innovation Projects (Grant ZG2023028), and the Ant Group.

APPENDIX

FORMAL PROOFS

Proof 1: We prove the EUF-aID-CMA security of our scheme via a game between the adversary $\mathcal{A}_1$ and challenger $\mathcal{C}$. If adversary $\mathcal{A}_1$ can forge the signcryptext with a non-negligible probability in PPT, then challenger $\mathcal{C}$ can use the forged signcryptext to solve the $\mathrm{SIS}_{q,n,2m,\beta}$ ($\beta = \sqrt{(4\sigma_s\sqrt{m}+2\sigma\kappa\sqrt{m}+2\kappa\sqrt{m})^2+(4\sigma_s\sqrt{m}+4\sigma\kappa\sqrt{m})^2}$) problem with a non-negligible probability.

Challenger $\mathcal{C}$ first built the following six lists.

L_1 (vehicle's identity ID, pseudonym $PID = (PID^1, PID^2)$);

L_2 (pseudonym PID or id, public key $\boldsymbol{P}_{PID(id)}$, H_3 hash value $\boldsymbol{H}_{PID(id)}$, PPK $\boldsymbol{D}_{PID(id)}$, SV $\boldsymbol{C}_{PID(id)}$);

L_3 (vehicle's identity ID_i, RSK r_{ID_i}, RP $f(x)$);

L_4 (pseudonym PID, message $\boldsymbol{\varpi}$, ($\boldsymbol{y}_{PID,1}$, $\boldsymbol{y}_{PID,2}$), current timestamp $\boldsymbol{t}_{PID}$, H_2 hash value $\boldsymbol{c}_{PID}$);

L_5 (random message $\boldsymbol{\tau}$, H_4 hash value $H_4(\boldsymbol{\tau})$);

L_6 (pseudonym PID, message $\boldsymbol{\varpi}$, RSU identity set $\boldsymbol{id} = (id_1, \cdots, id_l)$, current timestamp $\boldsymbol{t}_{PID}$, signcryptext $\boldsymbol{\varphi}_{PID}$).

1) Setup

Given the security parameter λ, Challenger $\mathcal{C}$ needs to prepare the following parameters:

The matrix dimension n, the prime modulus q ($q \equiv 1 \pmod{2m}$), the LWR modulus q' ($\frac{q}{q'} \geq \sqrt{n}$ is an integer), $m = 2n\lceil \log q \rceil$, $\bar{m} = w = n\lceil \log q \rceil$, the challenge bound κ ($\binom{n}{\kappa} 2^{\kappa} \geq 2^{128}$), the SIS Gaussian parameter $\sigma \geq \sqrt{n \log q} \times \omega(\sqrt{\log n})$, the signature Gaussian parameter $\sigma_s = \alpha\|\boldsymbol{Cc}\| = 24\sigma\kappa\sqrt{m}$, and the LWE Gaussian parameter $\sigma_{lwe} = \frac{q}{2q'\sqrt{3}}$.

Next, the challenger $\mathcal{C}$ defines the anti-collision hash functions $H_1: \{0,1\}^* \to \{0,1\}^{|ID|}$, $H_2: \{0,1\}^* \to \{\boldsymbol{v}: \boldsymbol{v} \in \{-1,0,1\}^n, \|\boldsymbol{v}\|_1 \leq \kappa\}$, $H_3: \{0,1\}^* \to \mathbb{Z}_q^{n \times n}$, and $H_4: \{0,1\}^* \to \{0,1\}^{|\boldsymbol{\varpi}|}$, samples the matrix $\boldsymbol{A}_S \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, and computes $(\boldsymbol{A}_K, \boldsymbol{R}_K) \leftarrow \mathrm{GenTrap}(n, m, \bar{m}, q)$, where the MPK is $\boldsymbol{A}_K \in \mathbb{Z}_q^{n \times m}$ and MSK is $\boldsymbol{R}_K$ ($\|\widetilde{\boldsymbol{R}_K}\| \leq O(\sqrt{n \log q})$). Finally, Challenger $\mathcal{C}$ sends the public parameters to the adversary $\mathcal{A}_1$.

2) Queries

(1) The RSK query

Given the number p of vehicles, challenger $\mathcal{C}$ samples p RSKs $r_{ID_i} \xleftarrow{\$} \mathbb{R}_q$ and the PRSK $b \xleftarrow{\$} \mathbb{R}_q$, computes $\boldsymbol{A}_K\boldsymbol{B} = \bar{\boldsymbol{A}}$ ($\boldsymbol{B} = \mathrm{rot}(b)$) and $f(x) = \prod_{i=1}^{p}(x - r_{ID_i}) + b$, and publishes $f(x)$ as the RP and $\bar{\boldsymbol{A}}$ as the RPK. Finally, challenger $\mathcal{C}$ sends r_{ID_i} ($i \in [p]$) to adversary $\mathcal{A}_1$ and saves (ID_i ($i \in [p]$), r_{ID_i} ($i \in [p]$), $f(x)$) in the list L_3.

(2) The hash query

a) The H_1 query: Given the identity ID, challenger $\mathcal{C}$ first queries the list L_1. If ID already exists in the list L_1, then challenger $\mathcal{C}$ sends the corresponding $PID^2 \oplus ID$ to adversary $\mathcal{A}_1$. Otherwise, challenger $\mathcal{C}$ runs the corresponding algorithm to generate the corresponding information of ID in the list L_1, sends $PID^2 \oplus ID$ to adversary $\mathcal{A}_1$.

b) The H_2 query: Given (ID, $\boldsymbol{\varpi}$, $\boldsymbol{t}_{PID}$), challenger $\mathcal{C}$ first obtains PID from the list L_1 and queries the list L_4. If (PID, $\boldsymbol{\varpi}$, $\boldsymbol{t}_{PID}$) already exists in the list L_4, then challenger $\mathcal{C}$ sends the corresponding $\boldsymbol{c}_{PID}$ to adversary $\mathcal{A}_1$. Otherwise, challenger $\mathcal{C}$ samples $\boldsymbol{y}_{PID,1} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^m,\sigma_s}$ and $\boldsymbol{y}_{PID,2} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^m,\sigma_s}$, and computes $\boldsymbol{c}_{PID}$. Finally, challenger $\mathcal{C}$ sends $\boldsymbol{c}_{PID}$ to adversary $\mathcal{A}_1$ and saves (PID, $\boldsymbol{\varpi}$, ($\boldsymbol{y}_{PID,1}$, $\boldsymbol{y}_{PID,2}$), $\boldsymbol{t}_{PID}$, $\boldsymbol{c}_{PID}$) in the list L_4.

c) The H_3 query: Given the identity $ID(id)$, challenger $\mathcal{C}$ first obtains PID from the list L_1 (if this is the vehicle's identity) and queries list L_2. If $PID(id)$ already exists in the list L_2, then challenger $\mathcal{C}$ sends the

corresponding $\boldsymbol{H}_{PID(id)}$ to adversary $\mathcal{A}_1$. Otherwise, challenger $\mathcal{C}$ runs the corresponding algorithm to generate the corresponding information of $PID(id)$ in the list L_2, sends $\boldsymbol{H}_{PID(id)}$ to adversary $\mathcal{A}_1$.

d) The H_4 query: Given the random message $\boldsymbol{\tau}$, challenger $\mathcal{C}$ first queries the list L_5. If $\boldsymbol{\tau}$ already exists in the list L_5, then challenger $\mathcal{C}$ sends the corresponding $H_4(\boldsymbol{\tau})$ to adversary $\mathcal{A}_1$. Otherwise, challenger $\mathcal{C}$ runs the corresponding algorithm to generate the corresponding information of $\boldsymbol{\varpi}$ in the list L_5, sends $H_4(\boldsymbol{\tau})$ to adversary $\mathcal{A}_1$.

(3) The PPK query

Given the identity $ID(id)$, challenger $\mathcal{C}$ first obtains PID from the list L_1 (if this is the vehicle's identity) and queries list L_2. If identity $PID(id)$ already exists in the list L_2, then challenger $\mathcal{C}$ sends the corresponding PPK $\boldsymbol{D}_{PID(id)}$ to adversary $\mathcal{A}_1$. Otherwise, challenger $\mathcal{C}$ runs the corresponding algorithm to generate the corresponding information of $PID(id)$ in the list L_2 and sends the PPK $\boldsymbol{D}_{PID(id)}$ to adversary $\mathcal{A}_1$.

(4) The SV query

Because the type I adversary can replace the signer's public key, the adversary $\mathcal{A}_1$ can perform the following two SV queries.

a) Do not replace the public key. Given the identity $ID(id)$, challenger $\mathcal{C}$ first obtains PID from the list L_1 (if this is the vehicle's identity) and queries list L_2. If $PID(id)$ already exists in the list L_2, then challenger $\mathcal{C}$ sends the corresponding $\boldsymbol{C}_{PID(id)}$ to adversary $\mathcal{A}_1$. Otherwise, challenger $\mathcal{C}$ runs the corresponding algorithm to generate the corresponding information of $PID(id)$ in the list L_2 and sends $\boldsymbol{C}_{PID(id)}$ to adversary $\mathcal{A}_1$.

b) Replace the public key. Given $(PID(id), \boldsymbol{P}_{PID(id)}, \boldsymbol{C}_{PID(id)})$ ($\boldsymbol{P}_{PID(id)}$ and $\boldsymbol{C}_{PID(id)}$ are used to replace $PID(id)$'s public key and the SV), challenger $\mathcal{C}$ first queries the list L_2, and performs the following different operations according to different situations:

(a) If $PID(id)$ exists in the list L_2, and the corresponding public key is equal to $\boldsymbol{P}_{PID(id)}$, then challenger $\mathcal{C}$ sends $\boldsymbol{C}_{PID(id)}$ to adversary $\mathcal{A}_1$.

(b) If $PID(id)$ exists in the list L_2, but the public key of $PID(id)$ in the list L_2 is not equal to $\boldsymbol{P}_{PID(id)}$, then the corresponding public key and SV are modified to $\boldsymbol{P}_{PID(id)}$ and $\boldsymbol{C}_{PID(id)}$ respectively. Finally, challenger $\mathcal{C}$ sends $\boldsymbol{C}_{PID(id)}$ to adversary $\mathcal{A}_1$.

(c) If $PID(id)$ does not exist in the list L_2, challenger $\mathcal{C}$ samples $\boldsymbol{H}_{PID(id)} \in \mathbb{Z}_q^{n\times n}$, and runs algorithm $\mathrm{SampleMat}(\boldsymbol{A}_K, \boldsymbol{R}_K, \boldsymbol{H}_{PID(id)}, \sigma)$ to generate the PPK $\boldsymbol{D}_{PID(id)}$. Next challenger $\mathcal{C}$ stores $(PID(id), \boldsymbol{P}_{PID(id)}, \boldsymbol{H}_{PID(id)}, \boldsymbol{D}_{PID(id)}, \boldsymbol{C}_{PID(id)})$ in the list L_2, and sends $\boldsymbol{C}_{PID(id)}$ to adversary $\mathcal{A}_1$.

(5) The signcryptext query

Given $(ID, \boldsymbol{\varpi}, \boldsymbol{id} = (id_1, \cdots, id_l), \boldsymbol{t}_{PID})$, challenger $\mathcal{C}$ first obtains PID from the list L_1 and queries list L_6. If PID's signcryptext on $(\boldsymbol{\varpi}, \boldsymbol{id}, \boldsymbol{t}_{PID})$ already exists in the list L_6, then challenger $\mathcal{C}$ sends the signcryptext $\boldsymbol{\varphi}_{PID}$ to adversary $\mathcal{A}_1$. Otherwise, challenger $\mathcal{C}$ runs the corresponding algorithm to generate the corresponding information of $(PID, \boldsymbol{\varpi}, \boldsymbol{id}, \boldsymbol{t}_{PID})$ in the list L_2, list L_3, list L_4, and list L_5. Next challenger $\mathcal{C}$ computes as follows:

a) Compute $\boldsymbol{z}_{PID,1} = (\boldsymbol{D}_{PID} + \boldsymbol{B})\boldsymbol{c}_{PID} + \boldsymbol{y}_{PID,1} \in \mathcal{D}_{\mathbb{Z}^m,\sigma_s}$ and $\boldsymbol{z}_{PID,2} = \boldsymbol{C}_{PID}\boldsymbol{c}_{PID} + \boldsymbol{y}_{PID,2} \in \mathcal{D}_{\mathbb{Z}^m,\sigma_s}$.

b) Output signature $(\boldsymbol{z}_{PID,1}, \boldsymbol{z}_{PID,2}, \boldsymbol{c}_{PID})$ with the probability of $\min(1, \frac{\mathcal{D}_{\mathbb{Z}^m,\sigma_s}(\boldsymbol{z}_{PID,1})}{M_1\mathcal{D}_{\mathbb{Z}^m,\sigma_s,(\boldsymbol{D}_{PID}+\boldsymbol{B})\boldsymbol{c}_{PID}}(\boldsymbol{z}_{PID,1})}, \frac{\mathcal{D}_{\mathbb{Z}^m,\sigma_s}(\boldsymbol{z}_{PID,2})}{M_2\mathcal{D}_{\mathbb{Z}^m,\sigma_s,\boldsymbol{c}_{PID}\boldsymbol{c}_{PID}}(\boldsymbol{z}_{PID,2})})$ ($M_1 \approx 3$, $M_2 \approx 3$, $M = \max(M_1, M_2)$).

c) Sample $\boldsymbol{r} \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^n,\sigma_{lwe}}$, compute $\boldsymbol{x} = \boldsymbol{\varpi} \oplus H_5(\boldsymbol{\tau})$, $\overline{\boldsymbol{P}}_{\boldsymbol{id}} = \sum_{id_i\in\boldsymbol{id}}(\boldsymbol{H}_{id_i} + \boldsymbol{P}_{id_i}) \bmod q$, $\boldsymbol{c}_{\boldsymbol{id},1} = \left\lfloor \frac{q'}{q}(\boldsymbol{A}_K{}^T\boldsymbol{r} \bmod q) \right\rfloor$, $\boldsymbol{c}_{\boldsymbol{id},2} = \lfloor \frac{q'}{q}(\boldsymbol{A}_S{}^T\boldsymbol{r} \bmod q) \rceil$, and $\boldsymbol{c}_{\boldsymbol{id},3} = \lfloor \frac{q'}{q}(\overline{\boldsymbol{P}}_{\boldsymbol{id}}{}^T\boldsymbol{r} \bmod q) \rceil + \left\lfloor \frac{q'}{2} \right\rfloor \boldsymbol{\tau} \bmod q'$, send $\boldsymbol{\varphi}_{PID} = (\boldsymbol{t}_{PID}, PID, \boldsymbol{z}_{PID,1}, \boldsymbol{z}_{PID,2}, \boldsymbol{c}_{PID}, \boldsymbol{x}, \boldsymbol{c}_{\boldsymbol{id},1}, \boldsymbol{c}_{\boldsymbol{id},2}, \boldsymbol{c}_{\boldsymbol{id},3}, \boldsymbol{id})$ to the adversary $\mathcal{A}_1$ and save $(PID, \boldsymbol{\varpi}, \boldsymbol{id}, \boldsymbol{t}_{PID}, \boldsymbol{\varphi}_{PID})$ in the list L_6.

3) The forgery

After the above queries are completed, the adversary $\mathcal{A}_1$ can forge a signcryptext $\boldsymbol{\varphi}_{PID^*} = (\boldsymbol{t}_{PID^*}, PID^*, \boldsymbol{z}_{PID^*,1}, \boldsymbol{z}_{PID^*,2}, \boldsymbol{c}_{PID^*}, \boldsymbol{x}, \boldsymbol{c}_{\boldsymbol{id}^*,1}, \boldsymbol{c}_{\boldsymbol{id}^*,2}, \boldsymbol{c}_{\boldsymbol{id}^*,3}, \boldsymbol{id}^*)$ on $(PID^*, \boldsymbol{\varpi}^*, \boldsymbol{t}_{PID^*}, \boldsymbol{id}^* = (id_1^*, \cdots, id_l^*))$ in PPT (PID^* had not been queried in the above PPK query and $(PID^*, \boldsymbol{\varpi}^*, \boldsymbol{t}_{PID^*}, \boldsymbol{id}^*)$ had not been queried in the above signcryptext query).

According to the General Forking Lemma [29], after Challenger $\mathcal{C}$ gets the forged signcryptext, and resamples the H_2 query value corresponding to $(PID^*, \boldsymbol{\varpi}^*, \boldsymbol{t}_{PID^*}, \boldsymbol{id}^*)$, adversary $\mathcal{A}_1$ can forge another signcryptext $\boldsymbol{\varphi}'_{PID^*} = (\boldsymbol{t}_{PID^*}, PID^*, \boldsymbol{z}'_{PID^*,1}, \boldsymbol{z}'_{PID^*,2}, \boldsymbol{c}'_{PID^*}, \boldsymbol{x}, \boldsymbol{c}_{\boldsymbol{id}^*,1}, \boldsymbol{c}_{\boldsymbol{id}^*,2}, \boldsymbol{c}_{\boldsymbol{id}^*,3}, \boldsymbol{id}^*)$ on $(PID^*, \boldsymbol{\varpi}^*, \boldsymbol{t}_{PID^*}, \boldsymbol{id}^*)$ with a non-negligible probability, so that

$$\begin{cases} \boldsymbol{A}_K\boldsymbol{z}_{PID^*,1} + \boldsymbol{A}_S\boldsymbol{z}_{PID^*,2} - (H_3(PID^*) + \boldsymbol{P}_{PID^*} + \overline{\boldsymbol{A}})\boldsymbol{c}_{PID^*} \\ = \boldsymbol{A}_K\boldsymbol{z}'_{PID^*,1} + \boldsymbol{A}_S\boldsymbol{z}'_{PID^*,2} - (H_3(PID^*) + \boldsymbol{P}_{PID^*} + \overline{\boldsymbol{A}})\boldsymbol{c}'_{PID^*} \\ \boldsymbol{c}_{PID^*} \neq \boldsymbol{c}'_{PID^*} \\ \boldsymbol{z}_{PID^*,1} \neq \boldsymbol{z}'_{PID^*,1} \\ \boldsymbol{z}_{PID^*,2} \neq \boldsymbol{z}'_{PID^*,2} \end{cases} . \quad (A.1)$$

Due to $\overline{\boldsymbol{A}} = \boldsymbol{A}_K\boldsymbol{B} \bmod q$, $H_3(PID^*) = \boldsymbol{A}_K\boldsymbol{D}_{PID^*} \bmod q$, and $\boldsymbol{P}_{PID^*} = \boldsymbol{A}_S\boldsymbol{C}_{PID^*} \bmod q$, we can get $\boldsymbol{A}_K\boldsymbol{z}_{PID^*,1} + \boldsymbol{A}_S\boldsymbol{z}_{PID^*,2} - (\boldsymbol{A}_K\boldsymbol{D}_{PID^*} + \boldsymbol{A}_S\boldsymbol{C}_{PID^*} + \boldsymbol{A}_K\boldsymbol{B})\boldsymbol{c}_{PID^*} \bmod q = \boldsymbol{A}_K\boldsymbol{z}'_{PID^*,1} + \boldsymbol{A}_S\boldsymbol{z}'_{PID^*,2} - (\boldsymbol{A}_K\boldsymbol{D}_{PID^*} + \boldsymbol{A}_S\boldsymbol{C}_{PID^*} + \boldsymbol{A}_K\boldsymbol{B})\boldsymbol{c}'_{PID^*} \bmod q$, $\boldsymbol{z}_{PID^*,1} \neq \boldsymbol{z}'_{PID^*,1}$, $\boldsymbol{z}_{PID^*,2} \neq \boldsymbol{z}'_{PID^*,2}$, $\boldsymbol{c}_{PID^*} \neq \boldsymbol{c}'_{PID^*}$. That is, $[\boldsymbol{A}_T|\boldsymbol{A}_K]\begin{bmatrix} \boldsymbol{z}_{PID^*,1} + \boldsymbol{D}_{PID^*}\boldsymbol{c}_{PID^*} + \boldsymbol{B}\boldsymbol{c}_{PID^*} - \boldsymbol{z}'_{PID^*,1} - \boldsymbol{D}_{PID^*}\boldsymbol{c}'_{PID^*} - \boldsymbol{B}\boldsymbol{c}'_{PID^*} \\ \boldsymbol{z}_{PID^*,2} + \boldsymbol{C}_{PID^*}\boldsymbol{c}_{PID^*} - \boldsymbol{z}'_{PID^*,2} - \boldsymbol{C}_{PID^*}\boldsymbol{c}'_{PID^*} \end{bmatrix} = \boldsymbol{0} \bmod q$, $\boldsymbol{z}_{PID^*,1} \neq \boldsymbol{z}'_{PID^*,1}$, $\boldsymbol{z}_{PID^*,2} \neq \boldsymbol{z}'_{PID^*,2}$, $\boldsymbol{c}_{PID^*} \neq \boldsymbol{c}'_{PID^*}$.

Due to $\|\boldsymbol{z}_{PID^*_1}\| \leq 2\sigma_s\sqrt{m}$, $\|\boldsymbol{z}_{PID^*_2}\| \leq 2\sigma_s\sqrt{m}$, $\|\boldsymbol{z}'_{PID^*_1}\| \leq 2\sigma_s\sqrt{m}$, $\|\boldsymbol{z}'_{PID^*_2}\| \leq 2\sigma_s\sqrt{m}$, $\|\boldsymbol{c}_{PID^*}\| \leq \sqrt{\kappa}$, $\|\boldsymbol{c}'_{PID^*}\| \leq \sqrt{\kappa}$, $\|\widetilde{\boldsymbol{D}_{PID^*}}\| \leq \sigma\sqrt{m}$, $\|\boldsymbol{C}_{PID^*}\| \leq 2\sigma\sqrt{m}$ and $\boldsymbol{B} \in \mathbb{Z}_2^{m\times n}$, then $\left\| \begin{matrix} \boldsymbol{z}_{PID^*,1} + \boldsymbol{D}_{PID^*}\boldsymbol{c}_{PID^*} + \boldsymbol{B}\boldsymbol{c}_{PID^*} - \boldsymbol{z}'_{PID^*,1} - \boldsymbol{D}_{PID^*}\boldsymbol{c}'_{PID^*} - \boldsymbol{B}\boldsymbol{c}'_{PID^*} \\ \boldsymbol{z}_{PID^*,2} + \boldsymbol{C}_{PID^*}\boldsymbol{c}_{PID^*} - \boldsymbol{z}'_{PID^*,2} - \boldsymbol{C}_{PID^*}\boldsymbol{c}'_{PID^*} \end{matrix} \right\|$

$$= \sqrt{\begin{matrix} \left\|\boldsymbol{z}_{PID^*,1} + \boldsymbol{D}_{PID^*}\boldsymbol{c}_{PID^*} + \boldsymbol{B}\boldsymbol{c}_{PID^*} - \boldsymbol{z}'_{PID^*,1} - \boldsymbol{D}_{PID^*}\boldsymbol{c}'_{PID^*} - \boldsymbol{B}\boldsymbol{c}'_{PID^*}\right\|^2 \\ + \left\|\boldsymbol{z}_{PID^*,2} + \boldsymbol{C}_{PID^*}\boldsymbol{c}_{PID^*} - \boldsymbol{z}'_{PID^*,2} - \boldsymbol{C}_{PID^*}\boldsymbol{c}'_{PID^*}\right\|^2 \end{matrix}}$$

$\leq \sqrt{(4\sigma_s\sqrt{m}+2\sigma\kappa\sqrt{m}+2\kappa\sqrt{m})^2+(4\sigma_s\sqrt{m}+4\sigma\kappa\sqrt{m})^2}$.

Next, we prove that $\left\| \begin{matrix} \boldsymbol{z}_{PID^*,1}+\boldsymbol{D}_{PID^*}\boldsymbol{c}_{PID^*}+\boldsymbol{B}\boldsymbol{c}_{PID^*}-\boldsymbol{z}'_{PID^*,1}-\boldsymbol{D}_{PID^*}\boldsymbol{c}'_{PID^*}-\boldsymbol{B}\boldsymbol{c}'_{PID^*} \\ \boldsymbol{z}_{PID^*,2}+\boldsymbol{C}_{PID^*}\boldsymbol{c}_{PID^*}-\boldsymbol{z}'_{PID^*,2}-\boldsymbol{C}_{PID^*}\boldsymbol{c}'_{PID^*} \end{matrix} \right\| \neq \boldsymbol{0}$. When the public key $\boldsymbol{P}_{PID^*}$ is generated by an adversary $\mathcal{A}_1$, adversary $\mathcal{A}_1$ can determine the SV $\boldsymbol{C}_{PID^*}$ and disclose it, then challenger $\mathcal{C}$ can determine the SV $\boldsymbol{C}_{PID^*}$. When the public key $\boldsymbol{P}_{PID^*}$ is generated by challenger $\mathcal{C}$, challenger $\mathcal{C}$ can determine the SV $\boldsymbol{C}_{PID^*}$. According to $m=2n\lceil \log q \rceil$ and Lemma 2, there is at least another PPK $\boldsymbol{D}'_{PID^*}$ with a probability of $1-2^{-100}$ satisfies $\boldsymbol{H}_{PID^*}=\boldsymbol{A}_K\boldsymbol{D}_{PID^*} \bmod q = \boldsymbol{A}_K\boldsymbol{D}'_{PID^*} \bmod q$ and $\boldsymbol{D}_{PID^*} \neq \boldsymbol{D}'_{PID^*}$. And challenger $\mathcal{C}$ cannot determine which PPK the adversary $\mathcal{A}_1$ will use in the forged signcryptext. Therefore, challenger $\mathcal{C}$ randomly selects a PPK that satisfies $\left\| \begin{matrix} \boldsymbol{z}_{PID^*,1}+\boldsymbol{D}_{PID^*}\boldsymbol{c}_{PID^*}+\boldsymbol{B}\boldsymbol{c}_{PID^*}-\boldsymbol{z}'_{PID^*,1}-\boldsymbol{D}_{PID^*}\boldsymbol{c}'_{PID^*}-\boldsymbol{B}\boldsymbol{c}'_{PID^*} \\ \boldsymbol{z}_{PID^*,2}+\boldsymbol{C}_{PID^*}\boldsymbol{c}_{PID^*}-\boldsymbol{z}'_{PID^*,2}-\boldsymbol{C}_{PID^*}\boldsymbol{c}'_{PID^*} \end{matrix} \right\| \neq \boldsymbol{0}$ with a probability of at least 1/2. Then challenger $\mathcal{C}$ can solve the $\mathrm{SIS}_{q,n,2m,\beta}$ ($\beta = \sqrt{(4\sigma_s\sqrt{m}+2\sigma\kappa\sqrt{m}+2\kappa\sqrt{m})^2+(4\sigma_s\sqrt{m}+4\sigma\kappa\sqrt{m})^2}$) problem with a non-negligible probability. □

Proof 2: Similar to the proof of Theorem 1, a game between the adversary $\mathcal{A}_2$ and challenger $\mathcal{C}$ is still used to prove that our scheme is EUF-aID-CMA. Assume that the adversary $\mathcal{A}_2$ can forge the signcryptext in PPT. Then the challenger $\mathcal{C}$ can use the forged signcryptext of the adversary $\mathcal{A}_2$ to solve the $\mathrm{SIS}_{q,n,2m,\beta}$ ($\beta = \sqrt{(4\sigma_s\sqrt{m}+2\sigma\kappa\sqrt{m}+2\kappa\sqrt{m})^2+(4\sigma_s\sqrt{m}+4\sigma\kappa\sqrt{m})^2}$) problem with a non-negligible probability. The adversary $\mathcal{A}_2$ first makes corresponding queries to the challenger $\mathcal{C}$ as in Theorem 1. By the assumption of the type II adversary, adversary $\mathcal{A}_2$ knows the MSK used in the system, but cannot replace the user's public key. After adversary $\mathcal{A}_2$'s queries are completed, the adversary $\mathcal{A}_2$ can forge a signcryptext $\boldsymbol{\varphi}_{PID^*}=(\boldsymbol{t}_{PID^*}, PID^*, \boldsymbol{z}_{PID^*,1}, \boldsymbol{z}_{PID^*,2}, \boldsymbol{c}_{PID^*}, \boldsymbol{x}, \boldsymbol{c}_{\boldsymbol{id}^*,1}, \boldsymbol{c}_{\boldsymbol{id}^*,2}, \boldsymbol{c}_{\boldsymbol{id}^*,3}, \boldsymbol{id}^*)$ on $(PID^*, \boldsymbol{\varpi}^*, \boldsymbol{t}_{PID^*}, \boldsymbol{id}^*=(id_1^*, \cdots, id_l^*))$ in PPT (PID^* had not been queried in the above SV query and $(PID^*, \boldsymbol{\varpi}^*, \boldsymbol{t}_{PID^*}, \boldsymbol{id}^*)$ had not been queried in the above signcryptext query).

After Challenger $\mathcal{C}$ gets the forged signcryptext, according to the General Forking Lemma [29], when challenger $\mathcal{C}$ resamples the H_2 query value corresponding to $(PID^*, \boldsymbol{\varpi}^*, \boldsymbol{t}_{PID^*}, \boldsymbol{id}^*=(id_1^*, \cdots, id_l^*))$, adversary $\mathcal{A}_2$ can forge another signcryptext $\boldsymbol{\varphi}'_{PID^*}=(\boldsymbol{t}_{PID^*}, PID^*, \boldsymbol{z}'_{PID^*,1}, \boldsymbol{z}'_{PID^*,2}, \boldsymbol{c}'_{PID^*}, \boldsymbol{x}, \boldsymbol{c}_{\boldsymbol{id}^*,1}, \boldsymbol{c}_{\boldsymbol{id}^*,2}, \boldsymbol{c}_{\boldsymbol{id}^*,3}, \boldsymbol{id}^*)$ on $(PID^*, \boldsymbol{\varpi}^*, \boldsymbol{t}_{PID^*}, \boldsymbol{id}^*)$ with a non-negligible probability, so that

$$\begin{cases} \boldsymbol{A}_K\boldsymbol{z}_{PID^*,1}+\boldsymbol{A}_S\boldsymbol{z}_{PID^*,2}-(H_3(PID^*)+\boldsymbol{P}_{PID^*}+\overline{\boldsymbol{A}})\boldsymbol{c}_{PID^*} \\ =\boldsymbol{A}_K\boldsymbol{z}'_{PID^*,1}+\boldsymbol{A}_S\boldsymbol{z}'_{PID^*,2}-(H_3(PID^*)+\boldsymbol{P}_{PID^*}+\overline{\boldsymbol{A}})\boldsymbol{c}'_{PID^*} \\ \boldsymbol{c}_{PID^*} \neq \boldsymbol{c}'_{PID^*} \\ \boldsymbol{z}_{PID^*,1} \neq \boldsymbol{z}'_{PID^*,1} \\ \boldsymbol{z}_{PID^*,2} \neq \boldsymbol{z}'_{PID^*,2} \end{cases} \quad (A.2)$$

Due to $\overline{\boldsymbol{A}}=\boldsymbol{A}_K\boldsymbol{B} \bmod q$, $H_3(PID^*)=\boldsymbol{A}_K\boldsymbol{D}_{PID^*} \bmod q$, and $\boldsymbol{P}_{PID^*}=\boldsymbol{A}_S\boldsymbol{C}_{PID^*} \bmod q$, we can get $\boldsymbol{A}_K\boldsymbol{z}_{PID^*,1}+\boldsymbol{A}_S\boldsymbol{z}_{PID^*,2}-(\boldsymbol{A}_K\boldsymbol{D}_{PID^*}+\boldsymbol{A}_S\boldsymbol{C}_{PID^*}+\boldsymbol{A}_K\boldsymbol{B})\boldsymbol{c}_{PID^*} \bmod q = \boldsymbol{A}_K\boldsymbol{z}'_{PID^*,1}+\boldsymbol{A}_S\boldsymbol{z}'_{PID^*,2}-(\boldsymbol{A}_K\boldsymbol{D}_{PID^*}+\boldsymbol{A}_S\boldsymbol{C}_{PID^*}+\boldsymbol{A}_K\boldsymbol{B})\boldsymbol{c}'_{PID^*} \bmod q$, $\boldsymbol{z}_{PID^*,1} \neq \boldsymbol{z}'_{PID^*,1}$, $\boldsymbol{z}_{PID^*,2} \neq \boldsymbol{z}'_{PID^*,2}$, $\boldsymbol{c}_{PID^*} \neq \boldsymbol{c}'_{PID^*}$. That is, $[\boldsymbol{A}_T|\boldsymbol{A}_S]\begin{bmatrix} \boldsymbol{z}_{PID^*,1}+\boldsymbol{D}_{PID^*}\boldsymbol{c}_{PID^*}+\boldsymbol{B}\boldsymbol{c}_{PID^*}-\boldsymbol{z}'_{PID^*,1}-\boldsymbol{D}_{PID^*}\boldsymbol{c}'_{PID^*}-\boldsymbol{B}\boldsymbol{c}'_{PID^*} \\ \boldsymbol{z}_{PID^*,2}+\boldsymbol{C}_{PID^*}\boldsymbol{c}_{PID^*}-\boldsymbol{z}'_{PID^*,2}-\boldsymbol{C}_{PID^*}\boldsymbol{c}'_{PID^*} \end{bmatrix} = \boldsymbol{0} \bmod q$, $\boldsymbol{z}_{PID^*,1} \neq \boldsymbol{z}'_{PID^*,1}$, $\boldsymbol{z}_{PID^*,2} \neq \boldsymbol{z}'_{PID^*,2}$, $\boldsymbol{c}_{PID^*} \neq \boldsymbol{c}'_{PID^*}$.

Due to $\|\boldsymbol{z}_{PID^*_1}\| \leq 2\sigma_s\sqrt{m}$, $\|\boldsymbol{z}_{PID^*_2}\| \leq 2\sigma_s\sqrt{m}$, $\|\boldsymbol{z}'_{PID^*_1}\| \leq 2\sigma_s\sqrt{m}$, $\|\boldsymbol{z}'_{PID^*_2}\| \leq 2\sigma_s\sqrt{m}$, $\|\boldsymbol{c}_{PID^*}\| \leq \sqrt{\kappa}$, $\|\boldsymbol{c}'_{PID^*}\| \leq \sqrt{\kappa}$, $\|\widetilde{\boldsymbol{D}_{PID^*}}\| \leq \sigma\sqrt{m}$, $\|\boldsymbol{C}_{PID^*}\| \leq 2\sigma\sqrt{m}$ and $\boldsymbol{B} \in \mathbb{Z}_2^{m\times n}$, then $\left\| \begin{matrix} \boldsymbol{z}_{PID^*,1}+\boldsymbol{D}_{PID^*}\boldsymbol{c}_{PID^*}+\boldsymbol{B}\boldsymbol{c}_{PID^*}-\boldsymbol{z}'_{PID^*,1}-\boldsymbol{D}_{PID^*}\boldsymbol{c}'_{PID^*}-\boldsymbol{B}\boldsymbol{c}'_{PID^*} \\ \boldsymbol{z}_{PID^*,2}+\boldsymbol{C}_{PID^*}\boldsymbol{c}_{PID^*}-\boldsymbol{z}'_{PID^*,2}-\boldsymbol{C}_{PID^*}\boldsymbol{c}'_{PID^*} \end{matrix} \right\|$

$= \sqrt{\begin{matrix} \left\|\boldsymbol{z}_{PID^*,1}+\boldsymbol{D}_{PID^*}\boldsymbol{c}_{PID^*}+\boldsymbol{B}\boldsymbol{c}_{PID^*}-\boldsymbol{z}'_{PID^*,1}-\boldsymbol{D}_{PID^*}\boldsymbol{c}'_{PID^*}-\boldsymbol{B}\boldsymbol{c}'_{PID^*}\right\|^2 \\ +\left\|\boldsymbol{z}_{PID^*,2}+\boldsymbol{C}_{PID^*}\boldsymbol{c}_{PID^*}-\boldsymbol{z}'_{PID^*,2}-\boldsymbol{C}_{PID^*}\boldsymbol{c}'_{PID^*}\right\|^2 \end{matrix}}$

$\leq \sqrt{(4\sigma_s\sqrt{m}+2\sigma\kappa\sqrt{m}+2\kappa\sqrt{m})^2+(4\sigma_s\sqrt{m}+4\sigma\kappa\sqrt{m})^2}$.

Next, we prove that $\left\| \begin{matrix} \boldsymbol{z}_{PID^*,1}+\boldsymbol{D}_{PID^*}\boldsymbol{c}_{PID^*}+\boldsymbol{B}\boldsymbol{c}_{PID^*}-\boldsymbol{z}'_{PID^*,1}-\boldsymbol{D}_{PID^*}\boldsymbol{c}'_{PID^*}-\boldsymbol{B}\boldsymbol{c}'_{PID^*} \\ \boldsymbol{z}_{PID^*,2}+\boldsymbol{C}_{PID^*}\boldsymbol{c}_{PID^*}-\boldsymbol{z}'_{PID^*,2}-\boldsymbol{C}_{PID^*}\boldsymbol{c}'_{PID^*} \end{matrix} \right\| \neq \boldsymbol{0}$. When the $\boldsymbol{H}_{PID^*}$ is generated by the adversary $\mathcal{A}_2$, adversary $\mathcal{A}_2$ can determine the PPK $\boldsymbol{D}_{PID^*}$ and disclose it, then challenger $\mathcal{C}$ can determine the PPK $\boldsymbol{D}_{PID^*}$. When the $\boldsymbol{H}_{PID^*}$ is generated by challenger $\mathcal{C}$, challenger $\mathcal{C}$ can determine the PPK $\boldsymbol{D}_{PID^*}$. By $m=2n\lceil \log q \rceil$ and Lemma 2, there exists another SV $\boldsymbol{C}_{PID^*}$ with probability $1-2^{-100}$ satisfying $\boldsymbol{P}_{PID^*}=\boldsymbol{A}_S\boldsymbol{C}_{PID^*} \bmod q = \boldsymbol{A}_S\boldsymbol{C}'_{PID^*} \bmod q$, $\boldsymbol{C}_{PID^*} \neq \boldsymbol{C}'_{PID^*}$. Therefore, challenger $\mathcal{C}$ cannot determine which SV the adversary $\mathcal{A}_2$ will use in the forged signcryptext and randomly selects an SV that satisfies $\left\| \begin{matrix} \boldsymbol{z}_{PID^*,1}+\boldsymbol{D}_{PID^*}\boldsymbol{c}_{PID^*}+\boldsymbol{B}\boldsymbol{c}_{PID^*}-\boldsymbol{z}'_{PID^*,1}-\boldsymbol{D}_{PID^*}\boldsymbol{c}'_{PID^*}-\boldsymbol{B}\boldsymbol{c}'_{PID^*} \\ \boldsymbol{z}_{PID^*,2}+\boldsymbol{C}_{PID^*}\boldsymbol{c}_{PID^*}-\boldsymbol{z}'_{PID^*,2}-\boldsymbol{C}_{PID^*}\boldsymbol{c}'_{PID^*} \end{matrix} \right\| \neq \boldsymbol{0}$ with a probability of at least 1/2. Furthermore, challenger $\mathcal{C}$ can solve the $\mathrm{SIS}_{q,n,2m,\beta}$ ($\beta = \sqrt{(4\sigma_s\sqrt{m}+2\sigma\kappa\sqrt{m}+2\kappa\sqrt{m})^2+(4\sigma_s\sqrt{m}+4\sigma\kappa\sqrt{m})^2}$) problem with a non-negligible probability. □

Proof 3:

1) Setup: The Setup phase is the same as the setup phase of Theorem 1.

2) Query phase 1: Except for the unsigncryption query, other queries are the same as those in Theorem 1.

3) Challenge: Adversary $\mathcal{A}_1$ samples two messages ($\boldsymbol{\varpi}_0$, $\boldsymbol{\varpi}_1$), the current timestamp $\boldsymbol{t}_{PID^*}$, and the identities (ID^*, PID^*) and $\boldsymbol{id}^*=(id_1^*, id_2^*, \cdots, id_l^*)$ (adversary $\mathcal{A}_1$ has not previously asked for the PPK query of $\boldsymbol{id}^*=(id_1^*, id_2^*, \cdots, id_l^*)$), and submits them to challenger $\mathcal{C}$. Next, challenger $\mathcal{C}$ samples a $c \xleftarrow{\$} \{0, 1\}$ and executes $\boldsymbol{\varphi}_{PID^*}=\mathrm{Signcryption}(ID^*, PID^*, \boldsymbol{\varpi}_c, \boldsymbol{t}_{PID^*}, \boldsymbol{id}^*)$. Then, challenger $\mathcal{C}$ sends $\boldsymbol{\varphi}_{PID^*}$ to adversary $\mathcal{A}_1$.

4) Query phase 2: Except for the PPK query about $\boldsymbol{id}^*$, which cannot be queried by the adversary $\mathcal{A}_1$, and the unsigncryption query about $\boldsymbol{\varphi}_{PID^*}$ cannot be queried by adversary $\mathcal{A}_1$, other queries are the same as the queries in Query phase 1.

Analysis and reduction argument:

Let the instance of the Decision SIS problem be ($\boldsymbol{A}_S$, $\boldsymbol{A}_K$, $\boldsymbol{Y}$) $\in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times n}$. The challenger $\mathcal{C}$'s goal is to determine whether $\boldsymbol{Y}$ is "real" (i.e., there exist short matrices

$\boldsymbol{C}_i$ and $\boldsymbol{D}_i$ such that $\boldsymbol{Y} = \boldsymbol{A}_S \sum_i \boldsymbol{C}_i + \boldsymbol{A}_K \sum_i \boldsymbol{D}_i \bmod q$) or "random" (i.e., uniformly distributed over $\mathbb{Z}_q^{n\times n}$).

During the simulation, challenger $\mathcal{C}$ uses the externally given ($\boldsymbol{A}_S$, $\boldsymbol{A}_K$) as part of the system public key and sets the aggregate public key for the challenge identity $\overline{\boldsymbol{P}}_{\boldsymbol{id}^*} = \boldsymbol{Y}$. According to Remark 5, challenger $\mathcal{C}$ uses this $\boldsymbol{Y}$ as the public key when generating the challenge signcryptext $\boldsymbol{\varphi}_{PID^*}$.

We now analyze the behavior of adversary $\mathcal{A}_1$ when it outputs its guess c' in two cases:

(1) Case real ($\boldsymbol{Y}$ is "real"):

If $\boldsymbol{Y}$ comes from a real instance of the Decision SIS problem, it is constructed as a linear combination of $\boldsymbol{A}_S$ and $\boldsymbol{A}_K$ with short matrices. In this case, due to the correctness of the scheme, the entire system environment, all oracle responses, and the challenge signcryptext simulated by challenger $\mathcal{C}$ using $\boldsymbol{Y}$ are computationally indistinguishable from the environment in the real IND-aID-CCA2 security experiment faced by adversary $\mathcal{A}_1$ (this relies on the perfect simulation of hash queries and private key extraction operations in the ROM, as shown in Theorem 1). Therefore, in case real, the game perceived by adversary $\mathcal{A}_1$ is consistent with the real game. By assumption, there exists a PPT adversary $\mathcal{A}_1$ that can break the IND-aID-CCA2 security of our scheme with a non-negligible advantage. Let its advantage be $\mathrm{Adv}_{\mathcal{A}_1}^{\mathrm{IND-aID-CCA2}}(\lambda) = \varepsilon(\lambda)$, where $\varepsilon(\lambda)$ is a non-negligible function of the security parameter λ. Consequently, in this case, the probability that adversary $\mathcal{A}_1$ correctly guesses the challenge bit c is:

$$\Pr[c' = c|\ \mathrm{Real}] = 1/2 + \varepsilon(\lambda). \quad \text{(A.3)}$$

(2) Case random ($\boldsymbol{Y}$ is "random"):

If $\boldsymbol{Y}$ is chosen uniformly at random from $\mathbb{Z}_q^{n\times n}$, then with overwhelming probability, $\boldsymbol{Y}$ is not a valid aggregate public key generated from $\boldsymbol{A}_S$ and $\boldsymbol{A}_K$. In the ROM, challenger $\mathcal{C}$ can program the random oracle such that even using a random $\boldsymbol{Y}$, it can generate a challenge signcryptext $\boldsymbol{\varphi}_{PID^*}$ that is statistically or computationally indistinguishable from a legitimate signcryptext. The crucial point is: Since $\boldsymbol{Y}$ is random, it does not correspond to any valid user private key (which should be able to unsigncrypt the signcryptext constructed based on $\boldsymbol{Y}$). In lattice-based cryptography, a signcryptext constructed based on a random matrix (or a matrix without a trapdoor) possesses statistical hiding properties, or the corresponding unsigncryption key information is perfectly hidden within the random oracle queries. Therefore, in this case, the information about the encrypted message $\boldsymbol{\varpi}_c$ in the challenge signcryptext $\boldsymbol{\varphi}_{PID^*}$ is completely hidden. The view of adversary $\mathcal{A}_1$ is statistically independent of the challenge bit c. Hence, its guess is essentially blind. Therefore, in this case, the probability that adversary $\mathcal{A}_1$ guesses correctly is at most:

$$\Pr[c' = c|\ \mathrm{Random}] \le 1/2 + \mathrm{negl}(\lambda), \text{ where } \mathrm{negl}(\lambda) \text{ is a negligible function.} \quad \text{(A.4)}$$

5) Guess: Adversary $\mathcal{A}_1$ outputs its guess c' for the challenge bit.

Decision by Challenger $\mathcal{C}$: Challenger $\mathcal{C}$ solves the given Decision SIS instance ($\boldsymbol{A}_S$, $\boldsymbol{A}_K$, $\boldsymbol{Y}$) by observing the output of adversary $\mathcal{A}_1$. Its decision algorithm is as follows:

If $c' = c$, then challenger $\mathcal{C}$ outputs "1", deciding that $\boldsymbol{Y}$ is "real" (i.e., from the SIS distribution).

If $c' \neq c$, then challenger $\mathcal{C}$ outputs "0", deciding that $\boldsymbol{Y}$ is "random".

Now, we calculate the advantage $\mathrm{Adv}_{\mathcal{C}}^{\mathrm{Deciosn-SIS}}(\lambda)$ of challenger $\mathcal{C}$ in solving the Decision SIS problem:

$$\begin{aligned}
&\mathrm{Adv}_{\mathcal{C}}^{\mathrm{Deciosn-SIS}}(\lambda)\\
&= |\Pr[\mathcal{C}\ \text{outputs 1}|\ \mathrm{Real}] - \Pr[\mathcal{C}\ \text{outputs 1}|\ \mathrm{Random}]|\\
&= |\Pr[c' = c|\ \mathrm{Real}] - \Pr[c' = c|\ \mathrm{Random}]|\\
&\ge |(1/2 + \varepsilon(\lambda)) - (1/2 + \mathrm{negl}(\lambda))|\\
&= |\varepsilon(\lambda) - \mathrm{negl}(\lambda)|\\
&\approx \varepsilon(\lambda)\ \text{(for sufficiently large } \lambda).
\end{aligned}$$

Since $\varepsilon(\lambda)$ is a non-negligible function, $\mathrm{Adv}_{\mathcal{C}}^{\mathrm{Deciosn-SIS}}(\lambda)$ is also a non-negligible function. This means that if there exists an IND-aID-CCA2 adversary $\mathcal{A}_1$ with a non-negligible advantage $\varepsilon(\lambda)$, then we have constructed an algorithm $\mathcal{C}$ that solves the Decision SIS problem with a non-negligible advantage (approximately $\varepsilon(\lambda)$). □

Proof 4:

As in [31], the proof of Theorem 4 proceeds in a similar fashion to that of Theorem 3. Even though the adversary $\mathcal{A}_2$ knows the MSK, we can still use a proof method similar to Theorem 3, based on the hardness of the Decision $\mathrm{SIS}_{q,n,m,\beta}$ problem, to prove that our scheme achieves IND-aID-CCA2 security; thus, the advantage of $\mathcal{A}_2$ is negligible. □

Remark 5: In the challenge phase in proof 3 and proof 4, the public key used by the challenger for signcryption is obtained from the oracle. It does not know whether it follows a uniform distribution $\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times n}$ or comes from the Decision $\mathrm{SIS}_{q,n,m,\beta}$ problem instance.

REFERENCES

[1] Z. Liu, J. Weng, J. Guo, J. Ma, F. Huang, H. Sun, and Y. Cheng, "PPTM: A privacy-preserving trust management scheme for emergency message dissemination in space–air–ground-integrated vehicular networks," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 5943–5956, 2022.

[2] J. Zhang, Y. Jiang, J. Cui, D. He, I. Bolodurina, and H. Zhong, "DBCPA: Dual blockchain-assisted conditional privacy-preserving authentication framework and protocol for vehicular ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 2, pp. 1127–1141, 2022.

[3] L. Wei, J. Cui, H. Zhong, Y. Xu, and L. Liu, "Proven secure tree based authenticated key agreement for securing V2V and V2I communications in VANETs," *IEEE Trans. Mobile Comput.*, vol. 21, no.9, pp. 3280–3297, 2022.

[4] H. Xiong, J. Chen, Q. Mei, and Y. Zhao, "Conditional privacy preserving authentication protocol with dynamic membership updating for VANETs," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 3, pp. 2089–2104, 2022.

[5] M. Al-Shareeda, S. Manickam, B. Mohammed, Z. Al-Mekhlafi, A. Qtaish, A. Alzahrani, G. Alshammari, A. Sallam, and K. Almekhlafi, "Provably secure with efficient data sharing scheme for fifth-generation (5G)-enabled vehicular networks without road-side unit (RSU)," *Sustainability*, vol. 14, no. 16, 2022.

[6] Y. Liang and Y. Liu, "Analysis and improvement of an efficient certificateless aggregate signature with conditional privacy preservation in VANETs," *IEEE Syst. J.*, vol. 17, no. 1, pp. 664–672, 2023.

[7] Y. Liang, H. Sun, X Zhang, Q. Xie, G. Liu, Z. Liu, "Lightweight Multifactor Authentication and Key Agreement Scheme in Vehicle-to-Grid Networks," *IEEE Trans. Veh. Technol.*, vol. 75, no. 4, pp. 6582-6599, 2025.

[8] Y. Liang, Y. Liu, and B. B. Gupta, "PPRP: Preserving-privacy route planning scheme in VANETs," *ACM Trans. Internet Technol.*, vol. 22, no. 4, pp. 1-18, 2022.

[9] H. Zhong, S. Zhang, J. Cui, L. Wei, and L. Liu, "Broadcast encryption

scheme for V2I communication in VANETs," *IEEE Trans. Veh. Technol.*, vol. 71, no. 3, pp. 2749–2760, 2022.

[10] Y. Yang, L. Zhang, Y. Zhao, K.-K.-R. Choo, and Y. Zhang, "Privacy preserving aggregation-authentication scheme for safety warning system in fog-cloud based VANET," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 317–331, 2022.

[11] M. Nielsen and I. Chuang, "Quantum computation and quantum information," *Cambridge Univ. Press*, Cambridge, pp. 558–559, 2000.

[12] P. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Rev.*, vol. 41, pp. 303–332, 1999.

[13] L. Grover, "A fast quantum mechanical algorithm for database search," *in: Proc. 28th Annu. ACM Symp. Theory Comput.*, 1996, pp. 212–219.

[14] M. Barbosa and P. Farshim, "Certificateless signcryption," *in: Proc. ACM Symp. Inf. Comput. Commun. Secur.*, 2008, pp. 369–372.

[15] I. Ullah, M. Khan, F. Khan, M. Jan, R. Srinivasan, and S. Mastorakis, "An efficient and secure multi-message and multi-receiver signcryption scheme for edge enabled Internet of vehicles," *IEEE Internet of Things J.*, vol. 9, no. 4, pp. 2688–2697, 2021.

[16] L. Wang, Z. Guan, Z. Chen, and M. Hu, "Multi-receiver signcryption scheme with multiple key generation centers through public channel in edge computing," *China Commun.*, vol. 19, no. 4, pp. 177–198, 2022.

[17] Y. Liang, H. Yan, and Y. Liu, "Unlinkable Signcryption Scheme for Multi-Receiver in VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 9, pp. 10138–10154, 2023.

[18] Y. Liang, G. Liu, X. Zhang, Y. Chen, J. Chen, Y. Xia, "Fully anonymous broadcast signcryption for secure health data transmission in WBANs," *IEEE Trans. Mob. Comput.*, vol. 25, no. 2, pp. 1729–1743, 2025.

[19] T. Su, G. Cheng, J. Huang, X. Zhao, S. Deng, "Secure and efficient personalized multi-receiver data sharing with cross-domain authentication for Internet of Vehicles," *IEEE Trans. Dependable Secure Comput.*, vol. 23, no.1, pp. 1608–1623, 2025.

[20] H. Le, D. Duong, P. Roy, W. Susilo, K. Fukushima, and S. Kiyomoto, "Lattice-based signcryption with equality test in standard model," *Comput. Stand. Interfaces*, vol. 76, 2021.

[21] H. Yu and J. Shi, "Certificateless multi-source signcryption with lattice," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 10, pp. 10157–10166, 2022.

[22] H. Yu and H. Wang, "Lattice-based threshold signcryption for blockchain oracle data transmission," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 10, pp. 11057-11065, 2023.

[23] M. Ajtai, "Generating hard instances of the short basis problem," *in Proc. Int. Colloq. Automata, Lang. Program.*, 1999, pp. 1–9.

[24] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *J. ACM*, vol. 56, no. 6, pp. 1-40, 2009.

[25] D. Micciancio and C. Peikert, "Trapdoors for lattices: simpler, tighter, faster, smaller," *in Proc. 31st Annu. Int. Conf. Theory Appl. Cryptogr. Techn.*, 2012, pp. 700–718.

[26] A. Banerjee, C. Peikert, and A. Rosen, "Pseudorandom functions and lattices," *in Proc. 31st Annu. Int. Conf. Theory Appl. Cryptogr. Techn.*, 2012, pp. 719–737.

[27] D. Micciancio and O. Regev, "Worst-case to average-case reductions based on Gaussian measures," *SIAM J. Comput.*, vol. 37, no. 1, pp. 267–302, 2007.

[28] V. Lyubashevsky, "Lattice signatures without trapdoors," *in Proc. 31st Annu. Int. Conf. Theory Appl. Cryptogr. Techn.*, 2012, pp. 738–755.

[29] M. Bellare, G. Neven, "Multi-signatures in the plain public-key model and a general forking lemma," *in Proc. 13th ACM Conf. Comput. Commun. Secur., Alexandria*, 2006, pp. 390–399.

[30] S. Y. Xu, X. Chen, Y. Guo, S. M. Yiu, S. Gao, and B. Xiao, "Efficient and secure post-quantum certificateless signcryption with linkability for IoMT," *IEEE Trans. Inf. Forensics Secur.*, vol. 20, pp. 1119-1134, 2025.

[31] Y. Bai, D. He, Z. Yang, M. Luo, and C. Peng, "efficient module-lattice-based certificateless online/offline signcryption scheme for Internet of Medical Things," *IEEE Internet Things J.*, vol, 12, no. 14, pp. 27350–27363, 2025.

[32] C. Cremers, "Scyther manual. Security protocol verification tool," GitHub repository, 2023. https://github.com/cascremers/scyther/blob/master/gui/scyther-manual.pdf (accessed 28 September 2025).

[33] Wikipedia Contributors, "Discrete Fourier Transform over a Ring—Number-Theoretic Transform," Wikipedia, The Free Encyclopedia, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Discrete_Fourier_transform_over_a_ring#Number-theoretic_transform (Accessed 28 September 2025).

[34] Wikipedia Contributors, "Exponentiation by squaring," Wikipedia, The Free Encyclopedia, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Exponentiation_by_squaring (Accessed: 28 Sept. 2025).

[35] D. Micciancio, A. Suhl, "Simulation-secure threshold PKE from LWE with polynomial modulus," *IACR Commun. Cryptol.*, vol. 01, no. 04, 2025.

[36] E. Fujisaki and T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes," J. Cryptol., vol. 26, pp. 80–101, 2013.

[37] The Sage Developers, "SageMath (Version 10.7)", 2025. [Online]. Available: https://www.sagemath.org

[38] PyCryptodome Developers, "SHAKE256," PyCryptodome 3.23.0 documentation, 2023. [Online]. Available: https://pycryptodome.readthedocs.io/en/latest/src/hash/shake256.html (Accessed: 28 Sept. 2025).

[39] Lattice Estimator Developers, "Lattice estimator 0.1 documentation," Read the Docs, 2021. [Online]. Available: https://lattice-estimator.readthedocs.io/en/latest/ (Accessed 28 September 2025).

[40] J. Anvers, A. Karmakar, and S. Roy, "Saber: module-lwr based key exchange, cpa-secure encryption and cca-secure kem," *in Proc. 10th Int. Conf. Cryptol. Afr.*, 2018, pp. 282–305.

[41] G. Cowan, "Lecture notes on data analysis," Universität Siegen, Siegen, Germany, Apr. 25, 1996.

[42] M. Sepulcre, M. Gonzalez-Martín, J. Gozalvez, R. Molina-Masegosa, and B. Coll-Perales, "Analytical models of the performance of IEEE 802.11p vehicle to vehicle communications," *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 713–724, 2022.

[43] C. Peikert, V. Vaikuntanathan, and B. Waters, "A framework for efficient and composable oblivious transfer," *in Proc. CRYPTO*, 2008, pp. 554–571.

[44] Z. Al-Mekhlafi, Z. Hanapi, M. Othman, and Z. Zukarnain, "A firefly-inspired scheme for energy-efficient transmission scheduling using a self-organizing method in a wireless sensor networks," *J. Comput. Sci.*, vol. 12, no. 10, pp. 482-494, 2016.

[45] Z. Al-Mekhlafi, Z. Hanapi, M. Othman, and Z. Zukarnain, "Travelling wave pulse coupled oscillator (twpco) using a self-organizing scheme for energy-efficient wireless sensor networks," *Plos One*, vol. 12, no. 1, 2017.

[46] Z. Al-Mekhlafi, Z. Hanapi, M. Othman, Z. Zukarnain, and A. Saleh, "Random traveling wave pulse-coupled oscillator algorithm of energy-efficient wireless sensor networks," *Int. J. Distrib. Sens. Netw.*, vol. 14, no. 4, 2018.

[47] Z. Al-Mekhlafi, Z. Hanapi, and A. Saleh, "Firefly-inspired time synchronization mechanism for self-organizing energy-efficient wireless sensor networks: A survey," *IEEE Access*, vol. 7, pp. 115229-115248, 2019.

[48] Z. Al-Mekhlafi, "Software-defined vehicular networks (sdvn)," *Int. J. Comput. Sci. Netw. Secur.*, vol. 22, no. 9, pp. 231-243, 2022.

[49] Z. Al-Mekhlafi and S. Alfhaid, "Innovative security measures: a comprehensive framework for safeguarding the internet of things," *in AI-Driven: Social Media Analytics and Cybersecurity*, Cham, Switzerland: Springer Nature, 2025. pp. 175-185.

[50] R. Cramer, L. Ducas, and B. Wesolowski, "Short Stickelberger Class Relations and Application to Ideal-SVP," *in Proc. 36th Annu. Int. Conf. Theory Appl. Cryptogr. Techn.*, 2017, pp. 324–348.