Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

A Scheme for Distributed Vehicle Authentication and Revocation in Decentralized VANETs

SUJASH NASKAR[†], **CARLO BRUNETTA**[‡], **GERHARD HANCKE** [§], **TINGTING ZHANG** [†], **MIKAEL GIDLUND** [†]

[†]Mid Sweden University, 85230, Sundsvall, Sweden (e-mail: firstname.lastname@miun.se)
[‡]e-mail: brunocarletta@gmail.com
[§]City University of Hongkong, Hongkong, (e-mail: gp.hancke@cityu.edu.hk)

Corresponding author: Sujash Naskar.

This work was part of the project "IoT Testbäddar," financed by the European Regional Development Fund and Region Västernorrland.

ABSTRACT Vehicular Ad-Hoc Networks (VANETs) offer enhanced road safety, efficient traffic management, and improved vehicle connectivity while dealing with privacy and security challenges in public communication. In these networks, authentication mechanisms are mandatory to establish trust among communicating entities, such as vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I), without losing identity and location-based privacy. The prevailing conventional authentication mechanisms frequently depend on a centralized trust authority (CA) to ensure the mutual verifiability of transmitted messages. Nevertheless, in scenarios where the density of vehicles within the network is notably high, an overwhelming influx of authentication requests may result in a communication bottleneck at the CA, leading to a single point of failure. This paper proposes a novel distributed authentication scheme in a decentralized VANET with multiple independent CAs connected to multiple local inspectors to eliminate a single point of failure. Furthermore, prior solutions lack the capability to immediately revoke a disputed vehicle that is transmitting malicious messages in the network. In this regard, the proposed scheme also facilitates an immediate revocation of a disputed sender to prevent other vehicles from further receiving malicious messages. As vehicles share time-sensitive data for driving assistance, our scheme minimizes the computation and communication costs for V2I key sharing and direct V2V authenticated message sharing significantly compared to previously proposed schemes. Using comparatively lightweight elliptic curve cryptography and eliminating the direct involvement of CAs in the authentication process, we have reduced the overall delays and achieved a maximum of ≈ 3.9 times faster V2I authenticated key sharing, and a maximum of ≈ 7.5 times faster V2V message sharing compared to state-of-the-art bilinear pairing-based protocols. A comprehensive efficiency analysis validates our scheme's ability to outperform time-sensitive responses, such as sending and receiving an alert within nearly 4 milliseconds.

INDEX TERMS Vehicular Ad-Hoc Networks (VANETs); Single point of failure; Privacy-preserving authentication; Revocation, Security attacks on VANET; Elliptic Curve Digital Signatures (ECDSA).

I. INTRODUCTION

VEHICULAR Ad Hoc Networks (VANETs) represent a pivotal technology at the intersection of transportation and communication systems, designed to enhance road safety, traffic management, and overall vehicular connectivity. These networks enable vehicles to seamlessly communicate with each other (V2V) and with infrastructure elements (V2I) [1] by sharing critical information about real-time traffic conditions, road hazards, and emergency alerts. VANETs introduce many innovative applications, including enhanced navigation, entertainment services, and transport efficiency optimizations. However, both V2V and V2I communication uses a public channel [2]; therefore, the shared sensitive information must be protected from potential privacy and security attacks in the communication channel. VANETs apply authentication mechanisms, that are pivotal in establishing trust among vehicles and infrastructure components [3]. These mechanisms verify the

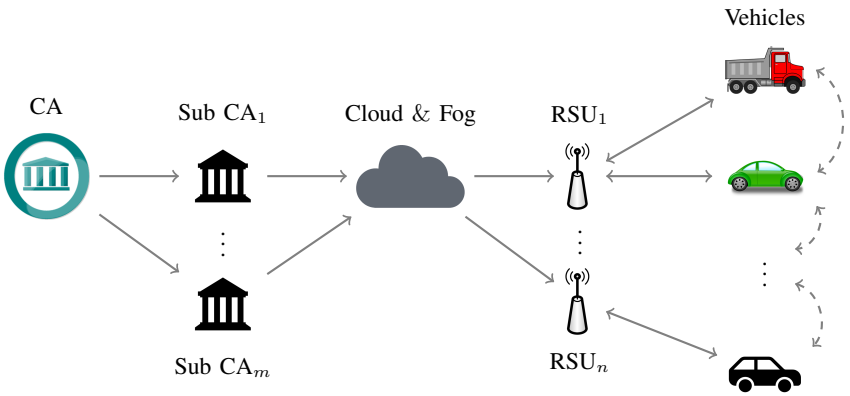

FIGURE 1: General Centralized VANET Architecture.

legitimacy of communicating participants, promising the information exchanged is reliable and untampered. Ensuring the confidentiality of sensitive data, prompt responses to time-critical events, and safeguarding against a spectrum of threats, including insider and outsider attacks on public channels [4], present formidable hurdles in vehicular communication. The effectiveness of VANETs in enhancing road safety and traffic management hinges on the designed robust authentication protocols that can withstand these unique challenges posed by adversaries in public channels.

In this context, the traditional centralized VANET architecture [5], as shown in Figure 1, comprises three key entities: a central trusted Certification Authority (CA), Roadside Units (RSUs) operating in conjunction with cloud and fog nodes [6], and vehicles equipped with tamper-protected On-Board Units (OBUs). These entities engage in wireless communication, typically governed by the CA, which assumes the role of a trusted entity responsible for all authentication verification and providing trust.

Despite its merits, the conventional centralized authentication models in VANETs suffer from a glaring vulnerability of potential communication bottleneck at the CA. With a high density of vehicles, a centralized authentication model in VANET with a single CA can get overloaded with authentication requests, which might lead to a single point of failure [7, 6]. In the event of a CA failure, the entire VANET communication ecosystem can collapse, with the CA struggling to manage an overwhelming workload and consequential response delays [8]. The involvement of RSUs further exacerbates delays in Vehicle-to-Vehicle (V2V) or Vehicle-to-Infrastructure (V2I) communication [5].

To address this critical single-point failure issue, recent years have seen distributed solutions emerge, particularly those based on public key infrastructure (PKI) and blockchain technologies [9]. These solutions distribute trust management across multiple entities within the communication hierarchy, moving away from sole dependence on a central authority. PKI-based approaches [10] incorporate semi-trusted multi-layer fog nodes or cloud computing devices, which collaborate with the CA to distribute session keys following vehicle identity verification [11, 12]. However, these protocols often compromise vehicle location and identity-based privacy and incur significant delays. Devices rely on the root CA for real-time database updates, rendering them vulnerable to communication bottlenecks. Blockchain-based distributed VANET systems [13, 14] maintain vehicle trustworthiness through mutual voting mechanisms and a comprehensive record-keeping system [15] that determines message acceptance or rejection in V2V communication. While effective, these schemes are highly time-consuming and demand substantial computational and storage resources. Moreover, they struggle to meet the stringent requirements of time-sensitive responses, particularly in emergency situations. Furthermore, using the existing solutions, immediate revocation of malicious vehicles becomes very complex or simply not possible. Consequently, there is a pressing need to devise a truly decentralized VANET network that balances distributed robust authentication and revocation with low communication and computation costs. This gap in the existing literature emphasizes the necessity for innovation and improvement.

Motivated by the imperative and challenging privacy and security needs, we present a novel solution for a decentralized VANET authentication ecosystem incorporating multiple CAs and local inspectors within the communication hierarchy in this paper. The proposed distributed authentication scheme eliminates the communication bottleneck at CA and performs authenticated key sharing locally from any available inspector. Unlike centralized models, authentication requests from vehicles do not Apart from distributed authentication, the beauty of the proposed scheme relies on two key facts: first, the vehicle's ability to respond quickly in time-critical situations, and second, an immediate revocation technique allowing vehicles to report and identify malicious vehicles in a privacy-preserving manner. Our proposed scheme is built upon Public Key Infrastructure, allowing precise management of session-specific keys for all V2I communication and enabling direct message sharing be-

tween vehicles (V2V). Our design enhances user autonomy by offering the flexibility of selecting an initial CA (parent CA) during vehicle registration. Leveraging lightweight Elliptic Curve Cryptography (ECC) [16] and one-way hash functions, our scheme ensures direct privacy-preserving V2I and V2V verifiability with minimal communication and computation overhead.

A. CONTRIBUTIONS

The proposed solution is an important step towards achieving these objectives in the context of VANET security with the following contributions:

1) **Eliminating single point failure with decentralized CAs:** Our approach introduces multiple independent CAs sharing sensitive but non-private data. This allows flexible registration and seamless movement of vehicles without compromising identity privacy. Compared to existing distributed VANETs, our scheme does not suffer from root-CA failure issues or real-time database updates for each verification. This decentralization makes our scheme truly distributed. In comparison with centralized VANETs, our scheme can achieve high scalability and reliability by eliminating a single point of failure when the number of vehicles is significantly high in the system. This is because each authentication request gets verified in a region-specific manner, and therefore, even if the number of vehicles significantly increases, the scheme distributes the requests to local inspectors and avoids network congestion at any CA.
2) **Efficient V2I Authenticated Key Sharing :** In our system, each CA is equipped with multiple Local Inspectors (LIs) that can directly verify vehicles (V2I) during an epoch similar to a session and share an epoch key locally with a computation cost of $\approx 3.45ms$ only. Unlike existing PKI-based V2I key sharing, the proposed scheme operates key-sharing without the help of CA or any cloud or fog nodes, reducing overall third-party communication. LIs only contact the CA when there are updates to the system, disputes, or vehicle change regions.
3) **Time-sensitive confidential and non-confidential V2V broadcasts:** Our scheme provides both V2V non-confidential and confidential broadcasts, while the existing protocols only allow non-confidential V2V broadcasts. Confidential V2V broadcasts allow a vehicle to send messages that are only readable by the specified intended receiver vehicle with an execution cost of $\approx 4.69ms$. The proposed scheme enables vehicles to exchange non-confidential verifiable messages directly with each other with an execution time of $\approx 3.08ms$, bypassing LIs, CAs, or any trust parties. In comparative studies, we have shown that state-of-the-art pairing-based PKI schemes have at most ≈ 7.5 times longer delay than our proposed scheme, making the proposed scheme at most ≈ 7.5 times faster in V2V authenticated message sharing.
4) **Immediate Privacy-Preserving Revocation:** We design a swift, privacy-preserving revocation process for malicious vehicles based on dispute reports from legitimate ones. A local inspector conducts the first revocation immediately to stop vehicles from accepting messages from a malicious sender. Then, only the CA can reveal the malicious sender's original identity if necessary. This is unique in our scheme as, so far, a fully functional revocation process to protect vehicles from receiving malicious messages immediately was missing in the literature.

The remainder of this paper is structured as follows: Section II presents an in-depth literature review of state-of-the-art schemes with their advantages and limitations. Section III presents the system model, assumptions, and requirements to lay the groundwork for the proposed scheme presented in Section IV. A comprehensive analysis of privacy and security is presented in Section V by following the requirements specified in Section III. The scheme's performance is evaluated in Section VI in terms of communication overhead and computation costs. We conduct a comparative study with state-of-the-art schemes and the proposed scheme in Section VII. Finally, Section VIII concludes the paper with discussions and insights.

II. RELATED WORK

Waheed et al. [19] analyzed a distributed task coordination system by using regional RSUs together with boundary relay vehicles, which voluntarily execute the task of other vehicles in its communication range to improve resource utilization and minimize the number of RSUs. A similar approach by Ali et al. [20] based on fog computing for geographically distributed VANET has shown how reducing excessive third-party use both in V2I and V2V communication can reduce overall authentication delays. A multi-fog-based authentication architecture proposed by Gu et al. [5] reduces the overall delay by adapting vehicle to fog verifiability. However, the protocol is centralized and suffers from a communication bottleneck at the CA, and the location-based privacy of the vehicles is not preserved. A decentralized two-phase authentication architecture [8] for VANET is proposed by Yang et al. based on authentication delegation to encounter the single-point failure issue. The first phase is a mutually verifiable token sharing between edge nodes and the vehicle. The second phase uses the token to verify the vehicle at any edge node. However, to share V2I authentication tokens, the protocol uses all the connected edge nodes to communicate; also, it uses bilinear pairing operations, which are computationally heavy. Wei et al. in [7] has addressed the delay-sensitive applications in VANET and the single point failure issues for a centralized authenticator and proposed a multi-CA model for a fog-based VANET to solve it. They have also shown the inefficiency of using bi-linear cryptography for delay-sensitive applications and proposed

Properties \ Scheme	Our scheme	Wang et al.[11]	Cui et al.[17]	Yang et al.[8]	Feng et al.[12]	Sikarwar et al.[18]	Wei et al.[7]
Authentication Type	V2I+V2V	V2I+V2V	V2I	V2I+V2V	V2V	V2V	V2I
Direct Verifiability	Yes	Yes	No	No	Yes	Yes	No
Cryptographic approach	ECC	Pairing	ECC	Pairing	Pairing	Pairing	ECC
Delay / Latency	Low	Very high	Very high	High	Very high	High	High
Lightweight / Heavy	Lightweight	Heavy	Lightweight	Heavy	Heavy	Heavy	Lightweight
Single point failure	Safe	Vulnerable	Vulnerable	Vulnerable	Vulnerable	Vulnerable	Vulnerable

TABLE 1: Comparative studies presenting the scheme properties for state-of-the-art schemes and our proposed scheme.

an authenticated key agreement protocol using lightweight ECC solutions. However, the root CA is centralized in their solution, and all the system components must be registered to the central root CA, limiting the system's distributive nature. A previously proposed bi-linear based solution by Zhang et al. in [10] with the concept of sub-TAs connected to a central root CA faces similar challenges. Also, each authenticated key agreement process involves the RSU, fog node, and a sub-CA twice (sending and receiving), significantly increasing the delay and reducing the system's performance. The scheme proposed by Sikarwar et al. [18] only allows direct authentication using pseudonym pooling. However, pairing-based cryptography makes the schemes heavy regarding computation delays. A PKI-based scheme by Cui et al. [17] also uses pseudonym pooling to achieve V2I key sharing with RSUS and direct V2V message sharing. But pseudonym pooling requires a huge storage, and revocation becomes very inefficient.

A hybrid scheme using ECC-based crypto combined with blockchain is proposed by Li et al. [13] for distributed verifiable message transfer. However, using this scheme, every vehicle in the system must undergo a complex, repetitive registration process each time they change an RSU region. Also, after each message verification, the vehicle performs a time-consuming feedback procedure to evaluate blockchain-based trustworthiness. Inedjaren et al. [14] and A. Ghaleb et al. [21] proposed trust-based V2V message delivery system in a distributive manner. This trustworthiness is evaluated with a reputation mechanism achieved by maintaining a blockchain-based trust table in each vehicle [14] and intrusion detection system (IDS) [21]. Another trust evaluating decentralized authentication and session key distribution scheme proposed by Ma et al. [22] uses a blockchain-based list. These protocols are distributed by not using any trusted authority and allowing only vehicles to determine the trustworthiness. However, blockchain and IDS-based schemes are extremely time-consuming and require huge computational and storage requirements, making all these schemes inefficient for delay-sensitive V2V responses. Also, the message complexity for these protocols to manage a voting mechanism is very high $\approx \mathcal{O}(n^2)$. Table 1 shows a comprehensive review of related schemes; each scheme is evaluated based on its merits and shortcomings, considering specific attributes as properties. To maintain a fair comparison, we've limited our focus to protocols that employ distributed or partially distributed authentication methods using ECC or pairing-based cryptography. Consequently, we have excluded blockchain-based protocols from our efficiency analysis.

III. SYSTEM MODEL AND PRELIMINARIES

The proposed system architecture has three layers consisting of several entities with different communication roles. As shown in Figure 2, each layer is connected with the entities at the level below. In this layered communication hierarchy, the entities at a lower level communicate with the immediate upper level following a designed subprotocol. Below, we first discuss the VANET components, their roles, and then the subprotocols they follow.

- **The Scheme Authority** or the SA is a global standardization authority like the ICAO [23] responsible for publishing the public parameters used by the CA and the other components in the system.
- **Certification Authorities**, in our scheme, many independent CAs are associated with separate public key pairs. Each CA is responsible for initially registering vehicles and providing a temporary pseudoID. CAs are considered to be fully trusted authorities located region or country-specific. CAs can communicate with each other by only sharing sensitive but non-private data.
- **Local Inspectors** or LIs are locally placed units responsible for V2I authenticated key sharing and providing traffic data to local vehicles. Each CA can have multiple independent LIs connected to it. LIs are considered semi-trusted entities as they are curious about specific privacy thefts. Also, LIs receive dispute reports and perform revocations.
- **Vehicles** are equipped with tamper-protected hardware chips known as the onboard unit or OBU responsible for all security-related computations and communications. Vehicles communicate with each other and with infrastructure by sharing authenticated messages. Vehicles are not trusted, and therefore, authentication is mandatory.
- **Communication Channel** A V2V and V2I communication follows the dedicated short-range communication or DSRC standards such as the IEEE 802.11p, IEEE 802.11px or C-V2X [24, 25, 26, 27] etc. Note that we assume that each LI is connected to its corresponding CA and that the CAs are connected via an

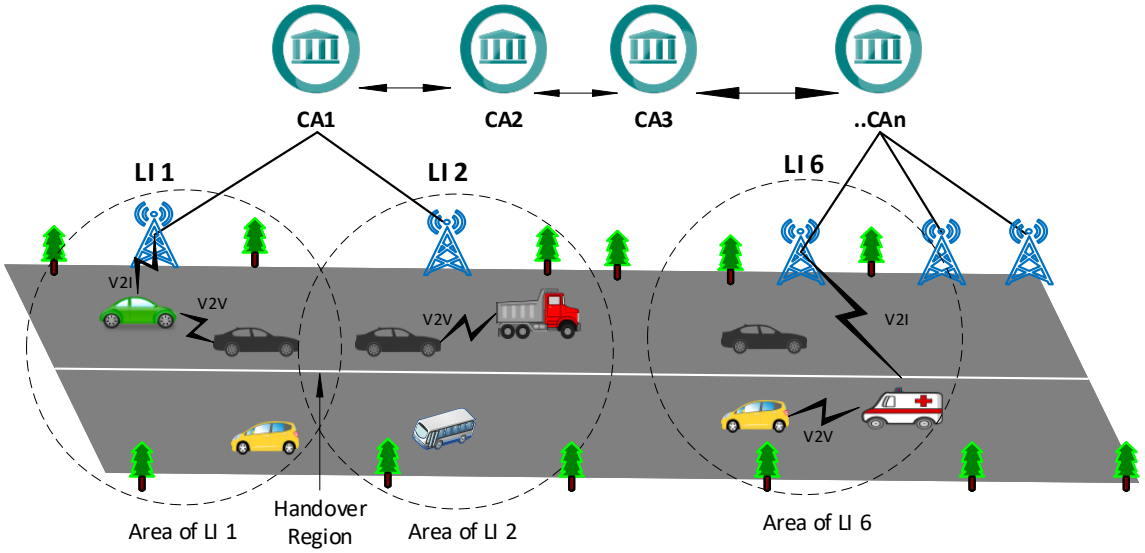

FIGURE 2: Proposed Distributed VANET Architecture

already established, authenticated, secure channel. This assumption is practical considering that CAs and LIs are trusted or semi-trusted and must communicate via a long-range communication channel such as a satellite or a long-distance 5G/4G network.

As presented in Figure 2, each LI has a public wireless coverage area where every vehicle can communicate with the LI. The handover region is where a vehicle switches communication from one LI to another. Each LI belongs to only one CA, and LIs do not communicate with each other directly. Vehicles use the public wireless channel to communicate with local LI and each other for traffic assistance. The communication between any LI and its CA takes place using the assumed pre-established secure channel.

A. THREAT MODEL

The proposed system follows a similar security threat model as mentioned in [28] and [4]. At first, all the entities in our system are separated into one of the following two categories.

- **Trusted Entities:** All the CAs and the scheme authority are assumed to be trusted entities. They do not pose any threat to the system and are secured against all possible attacks [7, 10].
- **Semi-Trusted Entities:** An LI in the system is assumed to be honest by not sharing secret information with any unauthorized entity or adversary. However, LIs might be curious to know the original identity of vehicles and their traffic paths, therefore threatening the privacy of vehicles. LIs in our system can also be called semi-trusted entities like RSUs, as mentioned in [12]. The OBU chip containing security-related secret information in a vehicle is assumed to keep data private. It is placed in a tamper-protected area of a vehicle to protect against physical attacks [23]. Even though a vehicle does not share its own secret information with others, it might be curious about traceability and identity-based privacy theft of other vehicles. It might also send false traffic information to create traffic hazards; therefore, vehicles are semi-trusted entities.

Moreover, any other entity that is not registered and external to the system is referred to as an unauthorized entity. Based on curious, semi-trusted, and unauthorized entities to the system, we now categorize the potential threat into internal and external adversaries as also mentioned in [4] by Zhang et al.

- **Internal Adversaries:** These entities are part of the designed VANET model; any dishonest vehicle and curious LI is an internal adversary to our system. They might be curious about privacy thefts such as the original identities, owner details, traffic routes, etc. Also, a malicious, dishonest vehicle can perform several security attacks, such as replay attacks, nonrepudiation, framing and sybil attacks in public communication.
- **External Adversaries:** Any unauthorized entity can threaten user privacy and try to perform both passive and active security attacks. These unauthorized entities are external adversaries to our system. In passive attacks, an external adversary monitors the public channel for any valuable information. In contrast, in active attacks, the adversary fabricates the messages and tries to enforce itself as a legal entity in the system.

We consider both internal and external adversaries to act as Dolev-Yao adversaries [29] with complete knowledge of the network structure, subprotocols, and huge computation

capabilities. The adversaries can intercept and read all messages exchanged among vehicles and between vehicles to LIs. Following the Dolev-Yao model, an adversary can try to modify or inject messages into communication and impersonate a legitimate vehicle or an LI.

B. PRIVACY AND SECURITY REQUIREMENTS

Our scheme's privacy and security requirements follow similar requirements as specified in [17, 12, 14, 28]. These requirements are generally adapted in almost every VANET communication as the basic security requirements.

- **No privacy leakage:** Ensuring that the original identity, traffic routes, and active and inactive timings of a vehicle are always private and secret from any adversary in communication.
- **Unlinkability:** Unlinkability of V2I authentication messages from any particular vehicle in two different instances protects the vehicle route from being traced in the network. However, during V2V broadcasts, messages by the exact vehicle should be linked by the receiver vehicles only for a short period, also known as an epoch, which is allowed for driving assistance and other traffic purposes.
- **Message Confidentiality and integrity:** Confidential messages containing sensitive data must not be known by anyone except the intended participants. Also, whenever a message is received, its integrity must be checked before the receiver accepts it.
- **Mutual verifiability:** Any two communicating entities, such as vehicles and local inspectors, must check each other's legal validity. Only a registered vehicle can be verified by the local inspector or by another vehicle during V2I or V2V data sharing. A malicious vehicle should never be able to pass a verification or forge itself as a verified vehicle to send messages successfully.
- **Accountability:** If a dispute is reported, the LI can revoke the misbehaving vehicle from the system by immediately putting it into the list of malicious vehicles.
- **Resistance against MITM:** The Man-in-the-middle attack (MITM) by an adversary listening to the communication should not succeed in acquiring secret or confidential data.
- **No replay attack:** Authentication credentials from one vehicle can not be replayed by itself or any adversary to get verified at any other time.
- **Impersonation/Framing free:** An adversary should not be able to impersonate a legal entity such as a registered vehicle or LI. Also, a sender's signature or identity can not be framed to send messages to a receiver.
- **Nonrepudiation:** The sender and receiver of a message in V2V direct communication can not deny sending and receiving messages in case of a dispute.
- **Sybil free:** Making sure that each vehicle is registered only once and they can not present multiple instances to confuse the system.
- **Reducing Denial of Service (DoS) attack:** If the network is flooded with malicious authentication requests, the designed protocol should be able to detect malicious requests quickly to become available to legitimate vehicles at all times.

C. SCHEME OVERVIEW

Considering the CAs as the top level in the hierarchy, the proposed three-layered VANET system performs the following subprotocols:

1) **Initialization:** Once SA has fixed the public security parameters, the CAs perform the initialization subprotocol to generate and publish their public keys.
2) **Registration:** Each vehicle's OBU needs to be registered by the CA before becoming a legitimate vehicle in the network. The driver's personal information and vehicle data are shared secretly with the CA using a secure channel in offline mode during this phase. The CA then generates a unique license for the vehicle and injects it with a temporary pseudoID into the vehicle OBU securely. Also, the local inspectors are initially registered with a corresponding CA.
3) **V2I authenticated key sharing:** In this phase, each vehicle in a particular region is verified by the LI and receives a secret symmetric key. This subprotocol is periodically performed to provide real-time traffic data and other information to legitimate active vehicles.
4) **V2V direct communication:** This subprotocol allows verified vehicles to broadcast authenticated messages directly to their surrounding vehicles. A receiver vehicle can then verify the message's authenticity locally. This communication typically includes sending emergency alerts, proximity alerts, or driving guidance.
5) **Revocation:** This subprotocol is designed to handle dispute reports. A vehicle can send a report about a misbehaving vehicle to the LI, which then can identify and revoke the misbehaving vehicle from the system. Only the CA can reveal the misbehaving vehicle's original identity if necessary.
6) **Handover:** When a vehicle changes from one LI to another, a handover protocol is performed between the vehicle and the new LI in a privacy-preserving manner that allows the vehicle to change to the new LI without being traced by the previous LI(s).

D. EPOCH AND EPOCH KEY

An epoch, ϵ, is a time frame similar to a session with a specified starting and ending time decided by the LIs. At the beginning of any epoch or during an epoch, vehicles in the network perform an authenticated V2I key sharing with the local verifier and receive a unique key valid only for that epoch. The local verifier decides the period of every epoch; however, the actual threshold is beyond our consideration in this paper. Once an epoch ϵ_1 ends, a new epoch ϵ_2 starts

with new authentication and pseudonyms for every active vehicle in the system.

With V2I authentication, each verified vehicle in the region of an LI gets a unique secret symmetric key called the epoch key (ek). The epoch key is used to perform AES (Advanced Encryption Standards) encryption and decryption on broadcast messages by the inspector or the vehicles. Once an epoch ends and a new one starts, the local inspector distributes a new epoch key with a V2I verification process to its region. Each epoch key is unique and unlikable for every session to guarantee both forward and backward security. An AES encryption is presented with the notation $\mathsf{ct} = \mathrm{Enc}\,(k, m)$, and decryption is presented as $m = \mathrm{Dec}\,(k, \mathsf{ct})$ where k, m, ct are a key, plaintext and cyphertext respectively.

E. ELLIPTIC CURVE DIGITAL SIGNATURE ALGORITHM (ECDSA)

For a prime field $\mathbb{F}_p$, where p is a large prime, an elliptic curve $\mathbb{E}$ is defined by the equation $\mathbb{E}_p(a, b) : y^2 = x^3 + ax + b \mod p$ with $a, b \in \mathbb{F}_p$. For a given point $P \in \mathbb{E}$, and any integer x, a scalar multiplication in ECC is given by $x \cdot P = P + P + ... + P(x\text{-times})$. Any point P with the smallest order q in ECC is called a base point if it can generate all the points in the curve, i.e., for q is the smallest positive integer for which $qP = O$, where O is the order of the elliptic curve. The security of ECC comes from the following properties:

- Elliptic curve discrete logarithm problem or ECDLP which states that for a given P and $Q = x.P$, it is computationally infeasible to find $x \in \mathbb{F}_p$ in polynomial time [16].
- Elliptic curve computational Defile-Hellman problem or ECCDHP states that given $P, Q = x.P, R = yP$, it is infeasible to compute xyP in a polynomial time.

Using the elliptic curve cryptographic principles, the ECDSA uses a public key pair $(\mathsf{sk}, \mathsf{pk} = \mathsf{sk}.P)$ to sign and verify signatures on messages [30]. A sender uses the secret key sk to sign the hashed value of a message m to generate the signature σ, an elliptic curve point. Then, the signature can be verified using the public key pk at the receiver side. The signature and verification process used in our protocol to sign and verify messages follows the below notations:
ECDSA signature Process: $\mathrm{Sign}\,(\mathsf{sk}, h(m)) \longrightarrow \sigma$, i.e. sign the hashed message $h(m)$ with secret key sk to generate signature σ.
ECDSA verification Process: $\mathrm{Ver}\,(\mathsf{pk}, h(m), \sigma) \longrightarrow (T/F)$, i.e. verify if the signature σ on hashed message $h(m)$ is valid (T) or invalid (F) using the public key pk.

IV. PROPOSED SCHEME

The proposed scheme is designed to achieve a mutual epoch-based V2I verification with key sharing, and V2V direct authenticated message sharing, suitable for the developed system and threat models mentioned above. The local inspector can authenticate a verification tuple sent by a vehicle during an epoch, and vehicles can use their public key to prove and communicate with other vehicles directly. A misbehaving vehicle can be reported to the inspector, who can then revoke it from the system without revealing its original identity. When a vehicle moves from one inspector region to another, a simple privacy-preserved handover protocol is performed for a smooth flow of the vehicle. The communication flow in all the protocol phases involving communicating parties and message transfer is denoted in Figure 3. All the notations used in the proposed scheme are presented with descriptions in Table 2. The detailed descriptions of the sub-protocols that construct our proposed model are presented below.

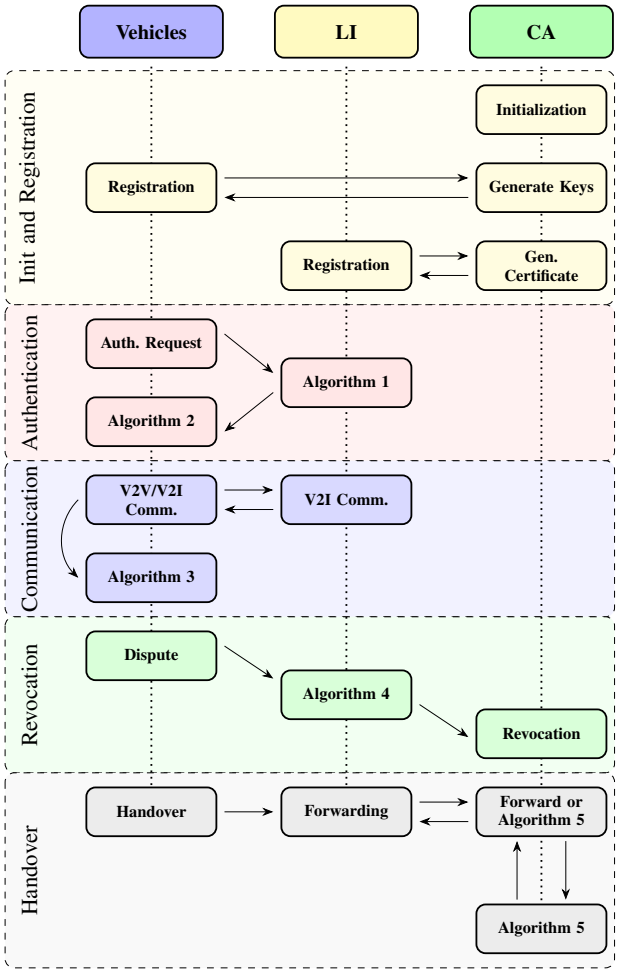

FIGURE 3: Protocols phases denoting the parties involved and the algorithms executed.

A. INITIALIZATION

At the very beginning, the scheme authority has to set all global security parameters and the elliptic curve $\mathbb{E}_p(a, b)$ over the finite field $\mathbb{F}_p$ with a base point P of order q in $\mathbb{E}_p$. Also, the SA decides three collision-free hash functions as below:

Notations	Definitions
P	Elliptic curve base point
σ	An ECDSA signature
$\delta_{\mathcal{V}}$	Randomized certificate of vehicle $\mathcal{V}$
$\mathsf{pk}_{\mathcal{C}}, \mathsf{sk}_{\mathcal{C}}$	Public and private key of a CA
$\mathsf{pk}_{\mathcal{V}}, \mathsf{sk}_{\mathcal{V}}$	Temporary public and private key of a vehicle
$\mathsf{pk}_{\mathcal{L}}, \mathsf{sk}_{\mathcal{L}}$	Public and private key of an LI
rk	Registration key of a CA
LIC	Unique license number of a vehicle
α	Long-term pseudoID of a vehicle
β	Ephemeral token
r	Auxiliary information
t	Timestamp
ek	An epoch key
ρ	Epoch specific value
$\mathrm{B-list}$	List of revoked vehicles
σ	An ECDSA signature
m, ct	Original message, encrypted message
$\mathsf{ct}_1, \mathsf{ct}_2$	Encrypted intermediate values
$\overline{m}$	Dispute summary
$\mathsf{ct}_{\overline{m}}$	Encrypted dispute summary with message m
$(\mathsf{pk}_{\mathcal{V}_S}, \mathsf{sk}_{\mathcal{V}_S})$	Public and private key of sender vehicle
$(\mathsf{pk}_{\mathcal{V}_R}, \mathsf{sk}_{\mathcal{V}_R})$	Public and Private key of receiver vehicle

TABLE 2: Notation Descriptions used in the Scheme.

1) An elliptic curve point to string hash function $\mathrm{H} : \mathbb{E}_p(a,b) \rightarrow \{0,1\}^l$ where l is the fixed size length of the hashed value.
2) A hash function to generate $k-bit$ fixed-length strings from random length message string as $\mathrm{H}_1 : \{0,1\}^* \rightarrow \{0,1\}^k$.
3) Finally, a hash function to generate field element from random length strings as $\mathrm{H}_2 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$.

Then, each CA is individually initialized with a public key pair by randomly selecting a secret value $\mathsf{sk}_{\mathcal{C}} \in \mathbb{Z}_q^*$ which serves as its private key and generates a public key as $\mathsf{pk}_{\mathcal{C}} = \mathsf{sk}_{\mathcal{C}} \cdot P$. Then, each CA publishes its own public key. The CAs use their public key pair to perform ECDSA on messages to generate verifiable signatures. Each CA is also initialized with a randomly generated secret string $(\mathsf{rk}_{\mathcal{C}})$ used as the vehicle registration key.

B. REGISTRATION

Once the initialization is completed, components such as local inspectors and vehicles can be added to the network by registering them to a corresponding CA using a secure channel. The registration process for a local inspector is as follows:

- The installation location of the LI, the network information that it is connected to, and a unique identity for each LI is decided and securely injected by a corresponding CA responsible for that specific region.
- The local inspector then chooses a random secret $\mathsf{sk}_{\mathcal{L}} \in \mathbb{Z}_q^*$ and computes its public key as $\mathsf{pk}_{\mathcal{L}} = \mathsf{sk}_{\mathcal{L}} \cdot P$
- Then the LI shares $\mathsf{pk}_{\mathcal{L}}$ with the CA, who signs the public key using ECDSA to generate the signature as $\mathrm{Sign}\left(\mathsf{sk}_{\mathcal{C}}, \mathrm{H}\left(\mathsf{pk}_{\mathcal{L}}\right)\right) \rightarrow \sigma$.
- Now CA shares $\left(\mathsf{pk}_{\mathcal{L}}, \sigma\right)$ to the corresponding inspector, who then can share it with vehicles within its region upon receiving a request.

A vehicle can initially register to a chosen CA, and the corresponding CA becomes a parent CA for that particular vehicle. The registration process for a vehicle with a corresponding CA is as follows:

- The vehicle shares vehicle information (VI), which may consist of vehicle serial number, manufacture information, and engine information, together with the owner's original identity (UI) to a regional CA using a secure channel.
- After receiving the information, the CA first computes $\mathrm{H}_1\left(\mathsf{VI}||\mathsf{UI}\right)$ and sends it to other connected CAs using a secure channel to check if the user is already registered to another CA. Other CAs can compute the same hash for their registered user and check if the received hash already exists. If it exists, the corresponding CA reports it to the requesting CA. Otherwise, it discards the message.
- Then the CA generates a unique license number LIC for the vehicle by computing $\mathsf{LIC} = \mathrm{H}_2\left(\mathsf{VI}||\mathsf{UI}||\mathsf{rk}_{\mathcal{C}}\right) \cdot P$.
- CA then generates a long term pseudoID $\alpha = \mathrm{H}\left(\mathrm{H}_2\left(\beta||\mathsf{t}\right) \cdot \mathsf{LIC}\right)$ where β is a random string that serves as an ephemeral token, and t is the current timestamp.
- CA stores the LIC permanently and stores corresponding α temporarily with an expiry time. Also, it injects the values $\{\mathsf{LIC}, \alpha, \beta\}$ and the expiry time in the OBU unit of the vehicle securely.
- The CA is now the parent CA for that particular vehicle, and it also shares $\{\alpha, \beta\}$ with the region-specific local inspector where the vehicle is issued using the pre-established channel.

C. AUTHENTICATION FOR V2I

To be able to use the VANET system resources and receive local traffic data, each vehicle must prove its authenticity to the local inspector. A vehicle first requests and receives the public key $(\mathsf{pk}_{\mathcal{L}})$ of the local inspector and then checks the validity of $\mathsf{pk}_{\mathcal{L}}$ using the ECDSA verification process as $\mathrm{Ver}\left(\mathsf{pk}_{\mathcal{C}}, \mathrm{H}\left(\mathsf{pk}_{\mathcal{L}}\right), \sigma\right) \rightarrow \{T, F\}$, where $\mathsf{pk}_{\mathcal{C}}$ is the public key of the corresponding CA under which the LI is registered. This verification takes place only once, and the vehicle accepts and stores the LI's public key for future use if the result is $'T'$. Following are the steps a vehicle performs that act as a prover during authentication with the local inspector as the verifier:

- the vehicle first selects a random value $\mathsf{sk}_{\mathcal{V}} \in \mathbb{Z}_q^*$ and computes its temporary public key as $\mathsf{pk}_{\mathcal{V}} = \mathsf{sk}_{\mathcal{V}} \cdot P$,
- then, it computes a randomized certificate $\delta_v = \mathrm{H}_1\left(\alpha||\mathsf{t}||\beta||\mathrm{H}\left(\mathsf{pk}_{\mathcal{V}}\right)\right)$ with current timestamp t,
- the vehicle computes an intermediate value $\mathsf{ct}_1 = \mathrm{Enc}\left(\mathrm{H}(\mathsf{sk}_{\mathcal{V}} \cdot \mathsf{pk}_{\mathcal{L}}), \beta||\mathsf{r}\right)$ with auxiliary information r, such as priority-based data or special requests that a vehicle might want to send. Priority data can be added

Algorithm 1: V2I Authentication Verification at LI

Input: Verification tuple $\{\delta_v, \mathsf{ct}_1, \mathsf{pk}_\mathcal{V}, \mathsf{t}\}$
Output: Verified with response tuple $[\mathsf{ack}, \mathsf{ct}_2, \mathsf{t}, \sigma_1]$ / Reject

if $(\mathsf{t}' - \mathsf{t}) \leq \Delta t$ **then**
  $(\beta||\mathsf{r}) = \mathrm{Dec}\left(\mathrm{H}(\mathsf{pk}_\mathcal{V} \cdot \mathsf{sk}_\mathcal{L}), \mathsf{ct}_1\right)$, extract β, r
  if $\mathrm{H}_1\left(\alpha||\mathsf{t}||\beta||\mathrm{H}\left(\mathsf{pk}_\mathcal{V}\right)\right) = \delta_v$ **then**
    Authentication successful
    Compute: $\mathsf{ct}_2 = \mathrm{Enc}\left(\mathrm{H}(\mathsf{pk}_\mathcal{V} \cdot \mathsf{sk}_\mathcal{L}), (\beta||\mathsf{ek}||\epsilon||\mathsf{r})\right)$ with new β value
    Sign the public key: $\mathrm{Sign}\left(\mathsf{sk}_\mathcal{L}, \mathrm{H}\left(\mathsf{pk}_\mathcal{V}\right)||\epsilon\right) \rightarrow \sigma_1$
    Compute $\mathsf{ack} = \mathrm{H}_1\left(\mathsf{ct}_2||\mathsf{t}||\alpha||\mathrm{H}\left(\sigma_1\right)\right)$ with current t
    Send response tuple: $[\mathsf{ack}, \mathsf{ct}_2, \mathsf{t}, \sigma_1]$
  else
    Authentication unsuccessful, reject
else
  Reject

by ambulance, police vehicle, or any other emergency response service vehicle that might need priority in the verification queue at LI.

- It then sends the authentication message tuple $\{\delta_v, \mathsf{ct}_1, \mathsf{pk}_\mathcal{V}, \mathsf{t}\}$ to the local inspector using the public channel.

In its current epoch, the local inspector generates and holds a unique epoch key ek, which will be shared with authenticated vehicles. Once a verification request is received at time t', the LI checks if $(\mathsf{t}' - \mathsf{t}) \leq \Delta t$, where Δt is a predefined threshold for the maximum allowed time between sending and receiving. If satisfied, the LI moves on to the following checks.

- Get back $(\beta||\mathsf{r}) = \mathrm{Dec}\left(\mathrm{H}(\mathsf{pk}_\mathcal{V} \cdot \mathsf{sk}_\mathcal{L}), \mathsf{ct}_1\right)$ and find a match with β in the local database of the LI to get corresponding α. If no match is found, LI discards the authentication request.
- If r is priority info and α belongs to a priority vehicle, put the authentication request at the top of the queue.
- Compute $\delta_v^{'} = \mathrm{H}_1\left(\alpha||\mathsf{t}||\beta||\mathrm{H}\left(\mathsf{pk}_\mathcal{V}\right)\right)$ and check if $\delta_v^{'} \stackrel{?}{=} \delta_v$. If yes, then the certificate is valid and authenticated. Otherwise, the certificate is invalid and rejected.
- Generate a new string β and compute intermediate $\mathsf{ct}_2 = \mathrm{Enc}\left(\mathrm{H}(\mathsf{pk}_\mathcal{V} \cdot \mathsf{sk}_\mathcal{L}), (\beta||\mathsf{ek}||\epsilon||\mathsf{r})\right)$ and r is an auxiliary value. Note r can also be a special response for priority vehicles, such as a group key or confidential message.
- Using ECDSA, sign the public key of the vehicle as $\mathrm{Sign}\left(\mathsf{sk}_\mathcal{L}, \mathrm{H}\left(\mathsf{pk}_\mathcal{V}\right)||\epsilon\right) \rightarrow \sigma_1$.
- Generate an acknowledgment as: $\mathsf{ack} = \mathrm{H}_1\left(\mathsf{ct}_2||\mathsf{t}||\alpha||\mathrm{H}\left(\sigma_1\right)\right)$, where t is the current timestamp.
- Send the authentication response tuple to the vehicle as $[\mathsf{ack}, \mathsf{ct}_2, \mathsf{t}, \sigma_1]$.
- Remove the old β value and replace it with the new one for the corresponding pseudoID α. This process is also presented in Algorithm 1.

Algorithm 2: V2I Authentication Acknowledgement at Vehicle

Input: Response tuple $[\mathsf{ack}, \mathsf{ct}_2, \mathsf{t}, \sigma_1]$
Output: Accept and store $(\beta, \mathsf{ek}, \epsilon)$/ Reject

if $(\mathsf{t}' - \mathsf{t}) \leq \Delta t$ **then**
  if $\mathrm{H}_1\left(\mathsf{ct}_2||\mathsf{t}||\alpha||\mathrm{H}\left(\sigma_1\right)\right) = \mathsf{ack}$ **then**
    $(\beta||\mathsf{ek}||\epsilon||\mathsf{r}) = \mathrm{Dec}\left(\mathrm{H}(\mathsf{sk}_\mathcal{V} \cdot \mathsf{pk}_\mathcal{L}), \mathsf{ct}_2\right)$
    Extract and store $\beta, \mathsf{ek}, \epsilon, \sigma_1$ for the epoch period
  else
    Reject and retry
else
  Reject and retry

Once received by the vehicle at time t', it checks if $(\mathsf{t}' - \mathsf{t}) \leq \Delta t$. If yes, then the vehicle performs the following computations to ensure that the confirmation came from the original local inspector and that the integrity of the message is preserved. This stepwise acknowledgment process is also presented in Algorithm 2.

- Compute and check if $\mathsf{ack}^{'} = \mathrm{H}_1\left(\mathsf{ct}_2||\mathsf{t}||\alpha||\mathrm{H}\left(\sigma_1\right)\right) \stackrel{?}{=} \mathsf{ack}$.
- If yes, then $(\beta||\mathsf{ek}||\epsilon||\mathsf{r}) = \mathrm{Dec}\left(\mathrm{H}(\mathsf{sk}_\mathcal{V} \cdot \mathsf{pk}_\mathcal{L}), \mathsf{ct}_2\right)$ Extract and save β, the epoch key ek, and the epoch information ϵ in its OBU memory temporarily.
- Stores the signature σ_1 until the epoch ends.

Once authenticated by the local inspector, vehicles actively join the network and receive traffic-related broadcasts from the local inspector (V2I) using the shared epoch key ek. Symmetric encryption can be done on the messages using the epoch key ek and then signed by the LI as proof of authenticity. Below is a detailed description of how vehicles communicate with each other following a similar process.

D. DIRECT V2V COMMUNICATION

Throughout an epoch, vehicles directly communicate with their surrounding vehicle(s) through short-range V2V communication and share important traffic messages. This message sharing must be mutually verifiable and privacy-preserving for both the sender and receiver sides. Let us consider an example where a sender vehicle $\mathcal{V}_S$ wants to send a message to the receiver vehicle $\mathcal{V}_R$. To do so, the sender vehicle $\mathcal{V}_S$ has to convince its legitimacy to the receiver $\mathcal{V}_R$. Also, $\mathcal{V}_R$ has to ensure that the received message is not from a sender $\mathcal{V}_S$ who has been blacklisted in the current epoch time ϵ by the corresponding LI. For ease of understanding, the public and private key pairs for $\mathcal{V}_S$ and $\mathcal{V}_R$ are represented by $(\mathsf{pk}_{\mathcal{V}_S}, \mathsf{sk}_{\mathcal{V}_S})$ and $(\mathsf{pk}_{\mathcal{V}_R}, \mathsf{sk}_{\mathcal{V}_R})$

Algorithm 3: V2V Direct Verification at Receiver

Input: Sender's tuple $[\zeta, m_h, \sigma_1, \sigma_2, \mathsf{pk}_{\mathcal{V}_S}, \mathsf{t}]$, and ρ
Output: Receiver accept/reject m

Compute $\rho = \mathrm{H}_1(\mathsf{ek}||\epsilon||\mathsf{t})$ with received t
if $\mathsf{pk}_{\mathcal{V}_S} \subseteq$ B$-$*list* **then**
  Reject
else
  if $\mathrm{Ver}\left(\mathsf{pk}_{\mathcal{L}}, \mathrm{H}\left(\mathsf{pk}_{\mathcal{V}_S}\right)||\epsilon, \sigma_1\right) \rightarrow \{T\}$ **then**
    Public key $\mathsf{pk}_{\mathcal{V}}^{S}$ is valid in current epoch ϵ
    if $\mathrm{Ver}\left(\mathsf{pk}_{\mathcal{V}_S}, \mathrm{H}_1(\zeta||\rho||m_h), \sigma_2\right) \rightarrow \{T\}$ **then**
      Sender $\mathcal{V}_S$ indeed signed ζ
      if *receive* $=$ *non-confidential broadcast* **then**
        $m = \mathrm{Dec}(\mathsf{ek}, \zeta)$
      else
        if *receive* = *confidential* **then**
          $m = \mathrm{Dec}\left(\mathrm{H}(\mathsf{sk}_{\mathcal{V}_R} \cdot \mathsf{pk}_{\mathcal{V}_S}), \zeta\right)$
      if $\mathrm{H}_1(m) = m_h$ **then**
        Accept m
      else
        Reject m
    else
      Invalid signature and reject m
  else
    Invalid Public key and reject m

respectively. The following are the steps of the direct V2V communication sub-protocol:

- The sender vehicle $\mathcal{V}_S$ generates the message m with timestamp t and computes an epoch specific value $\rho = \mathrm{H}_1(\mathsf{ek}||\epsilon||\mathsf{t})$.
- Depending on the type of the message m, there are two cases; case 1: m is a broadcast message (not confidential to receiver vehicles), and case 2: m is a confidential message to receiver $\mathcal{V}_R$).
- For case 1, the sender $\mathcal{V}_S$ encrypts m using the epoch key as $\zeta = \mathrm{Enc}(\mathsf{ek}, m)$ and for case 2, the sender encrypts m as $\zeta = \mathrm{Enc}\left(H(\mathsf{sk}_{\mathcal{V}_s} \cdot \mathsf{pk}_{\mathcal{V}_R}), m\right)$.
- Then $\mathcal{V}_S$ computes a message digest $m_h = \mathrm{H}_1(m)$ and using its secret key it signs ζ, ρ, m_h as $\mathrm{Sign}(\mathsf{sk}_{\mathcal{V}_S}, \mathrm{H}_1(\zeta||\rho||m_h)) \rightarrow (\sigma_2)$.
- For a non-confidential broadcast (case 1), $\mathcal{V}_S$ sends the tuple $[\zeta, m_h, \sigma_1, \sigma_2, \mathsf{pk}_{\mathcal{V}_S}, \mathsf{t}]$.
- For a confidential message (case 2), the vehicle $\mathcal{V}_S$ includes the public key of the intended receiver $\mathsf{pk}_{\mathcal{V}_R}$ in the tuple and then sends it.

Once received, the receiver vehicle $\mathcal{V}_R$ checks the timestamp and verifies the following to accept the message m from $\mathcal{V}_S$. This verification process at the receiver vehicle is also presented in Algorithm 3.

- Receive only if LI has not blacklisted the sender vehicle's public key, i.e., reject the message if $\mathsf{pk}_{\mathcal{V}_S} \subseteq$ [B-list]; otherwise, accept and continue.
- Compute epoch-specific value $\rho = \mathrm{H}_1(\mathsf{ek}||\epsilon||\mathsf{t})$ using the received t.
- Check if the received public key of the sender $\mathsf{pk}_{\mathcal{V}_S}$ is signed by the local inspector in the current epoch using ECDSA as: $\mathrm{Ver}\left(\mathsf{pk}_{\mathcal{L}}, \mathrm{H}\left(\mathsf{pk}_{\mathcal{V}_S}\right)||\epsilon, \sigma_1\right) \rightarrow \{T, F\}$. If the signature is valid, then it confirms that the received public key is valid or otherwise malicious.
- Check that the received encrypted message ζ and message digest m_h is indeed signed by the corresponding $\mathcal{V}_S$ by using $\mathsf{pk}_{\mathcal{V}_S}$ such that $\mathrm{Ver}\left(\mathsf{pk}_{\mathcal{V}_S}, \mathrm{H}_1(\zeta||\rho||m_h), \sigma_2\right) \rightarrow \{T, F\}$. The receiver accepts the encrypted message ζ if the signature is valid. Otherwise, it rejects the message.
- If ζ is received as a broadcast from $\mathcal{V}_S$, then decrypt it using the epoch key as $m = \mathrm{Dec}(\mathsf{ek}, \zeta)$.
- Otherwise, if ζ is a unicast, receiver $\mathcal{V}_R$ decrypts it as $m = \mathrm{Dec}\left(\mathrm{H}(\mathsf{sk}_{\mathcal{V}_R} \cdot \mathsf{pk}_{\mathcal{V}_S}), \zeta\right)$.
- Finally, check if $\mathrm{H}_1(m_h) \stackrel{?}{=} m_h$. Accept only if they match or reject.

If all the above checks are passed, $\mathcal{V}_R$ accepts the message m from $\mathcal{V}_S$.

E. REVOCATION

Algorithm 4: Revocation by LI

Input: Report $[\sigma_2, \sigma_3, \zeta, m_h, \mathsf{t}, \mathsf{ct}_{\overline{m}}, \mathsf{pk}_{\mathcal{V}_S}, \mathsf{pk}_{\mathcal{V}_R}]$
Output: Revoke $\mathsf{pk}_{\mathcal{V}_S}$/ Account $\mathsf{pk}_{\mathcal{V}_R}$ / Malicious

Compute $\rho = \mathrm{H}_1(\mathsf{ek}||\epsilon||\mathsf{t})$ using received t
if $\mathrm{Ver}\left(\mathsf{pk}_{\mathcal{V}_S}, \mathrm{H}_1(\zeta||\rho||m_h), \sigma_2\right) \rightarrow \{T\}$ & $\mathrm{Ver}\left(\mathsf{pk}_{\mathcal{V}_R}, \mathrm{H}_1(\mathsf{ct}_{\overline{m}}||\rho), \sigma_3\right) \rightarrow \{T\}$ **then**
  $\mathrm{Dec}\left(\mathrm{H}(\mathsf{sk}_{\mathcal{L}}.\mathsf{pk}_{\mathcal{V}_R}), \mathsf{ct}_{\overline{m}}\right) = \overline{m}||m$
  if $m_h = \mathrm{H}_1(m)$ **then**
    Contect analysis of $m, \overline{m}$ (beyond our scope)
    Blacklist $\mathsf{pk}_{\mathcal{V}_S}$ and Revoke $\mathcal{V}_S$
    Notify the parent CA of $\mathcal{V}_S$
  else
    Message m is altered, $\mathcal{V}_R$ is accountable
else
  if $\mathrm{Ver}\left(\mathsf{pk}_{\mathcal{V}_S}, \mathrm{H}_1(\zeta||\rho||m_h), \sigma_2\right) \rightarrow \{F\}$ & $\mathrm{Ver}\left(\mathsf{pk}_{\mathcal{V}_R}, \mathrm{H}_1(\mathsf{ct}_{\overline{m}}||\rho), \sigma_3\right) \rightarrow \{T\}$ **then**
    Receiver $\mathcal{V}_R$ is accountable for framing
  else
    Malicious Report

Suppose a vehicle $\mathcal{V}_R$ wants to send a dispute report committed by another vehicle $\mathcal{V}_S$ to the LI. In that case, first $\mathcal{V}_R$ creates a message $\overline{m}$ that includes a dispute summary. Then it encrypts it as $\mathsf{ct}_{\overline{m}} = \mathrm{Enc}\left(\mathrm{H}(\mathsf{sk}_{\mathcal{V}_R}.\mathsf{pk}_{\mathcal{L}}), \overline{m}||m\right)$. Then, it signs $\mathsf{ct}_{\overline{m}}$ using its secret key as: $\mathrm{Sign}(\mathsf{sk}_{\mathcal{V}_R}, \mathrm{H}_1(\mathsf{ct}_{\overline{m}}||\rho)) \rightarrow \sigma_3$; where ρ is same as generated when receiving the message from $\mathcal{V}_S$. Finally, a dispute report tuple containing

$[\sigma_2, \sigma_3, \zeta, m_h, \mathsf{ct}_{\overline{m}}, \mathsf{pk}_{\mathcal{V}_S}, \mathsf{pk}_{\mathcal{V}_R}, \mathsf{t}]$ is sent to the local inspector. Note that the timestamp t in this tuple is the same as received from $\mathcal{V}_S$. Now, the local inspector performs the following steps:

- Compute $\rho = \mathrm{H}_1(\mathsf{ek}||\epsilon||\mathsf{t})$ with received t.
- Verify ζ was indeed signed by the disputed sender $\mathcal{V}_S$ as $\mathrm{Ver}\left(\mathsf{pk}_{\mathcal{V}_S}, \mathrm{H}_1(\zeta||\rho||m_h), \sigma_2\right) \rightarrow \{T, F\}$ and the incident report message is signed by $\mathcal{V}_R$ as $\mathrm{Ver}\left(\mathsf{pk}_{\mathcal{V}_R}, \mathrm{H}_1(\mathsf{ct}_{\overline{m}}||\rho), \sigma_3\right) \rightarrow \{T, F\}$.
- If ζ is not signed by $\mathcal{V}_S$, and incident report message $\mathsf{ct}_{\overline{m}}$ is signed by the receiver $\mathcal{V}_R$, the LI confirms that $\mathcal{V}_R$ is trying to send a false report by framing $\mathcal{V}_S$ and therefore $\mathcal{V}_R$ is accountable.
- If both signatures are valid (T), the LI can now decrypt $\mathsf{ct}_{\overline{m}}$ and get the disputed message m and the dispute summary $\overline{m}$.
- Then it checks if received $m_h \stackrel{?}{=} \mathrm{H}_1(m)$, to confirm that the disputed message m has not been altered by $\mathcal{V}_R$. If m is altered, $\mathcal{V}_R$ is accountable. If not altered, LI performs a message content check of m and $\overline{m}$ (beyond our scope) to decide whether or not to put the public key of $\mathcal{V}_S$ in B-list.
- If LI blacklists a disputed vehicle $\mathcal{V}_S$, it broadcasts the most recent B-list to its region using the epoch key and a signature (same as V2V). Then, it informs the parent CA of the corresponding vehicle $\mathcal{V}_S$ about being revoked. LI can reject V2I authenticated key sharing for the disputed vehicle for the next epoch.
- For any other case, the LI considers the report malicious and rejects it.

Algorithm 4 presents the stepwise revocation performed by the LI after receiving a report. Note that if a dispute is reported, the original identity of the malicious vehicle is only revealed to the corresponding CA and not to the LI, which can only block the vehicle by blacklisting and rejecting further authentication. This preserves the original identity-based privacy of the vehicle at LI. This is unique as it allows multi-level dispute management from a very minor to a major dispute. A legitimate vehicle can then update the most recent B-list received from the LI and cease receiving messages from the identified malicious senders listed in B-list. This ensures that vehicles do not accept messages from malicious senders, thereby enhancing their safety.

F. HANDOVER

When a vehicle $\mathcal{V}$ moves from one inspector's region to another, a handover sub-protocol is performed to facilitate a smooth and easy vehicle transition in the system. While switching from one LI region to another, the vehicle's traffic route-based privacy must be preserved. Algorithm 5 presents the handover protocol performed by an LI to which a vehicle has requested a handover; the detailed description is below.

There are two following cases of handover sub-protocol:

Algorithm 5: Handover at CA

Input: Handover tuple $[\mathrm{A}_f, \mathsf{pk}^*, \mathsf{pk}_{\mathcal{C}}, \sigma_3]$
Output: Share (α, β) with requesting LI/CA

if $\mathrm{Ver}\left(pk^*, \mathrm{H}_1(\mathrm{A}_f||\mathrm{H}(\mathsf{pk}^*)||\mathrm{H}(\mathsf{pk}_{\mathcal{C}})), \sigma_3\right) \longrightarrow \{T\}$ **then**
  Compute $\mathrm{Dec}\left(\mathrm{H}(\mathsf{pk}_{\mathcal{V}} \cdot \mathsf{sk}_{\mathcal{C}}), \mathrm{A}_f\right)$
  Extract α, α_1, β and t
  if $\alpha \in record \ \ \& \ \ \alpha \nsubseteq B - list$ **then**
    Compute $\alpha_1^* = \mathrm{H}(\mathsf{LIC} \cdot \mathrm{H}_2(\beta||\mathsf{t}))$
    if $\alpha_1^* = \alpha_1$ **then**
      Share (α_1, β) to requesting LI/CA
    else
      Malicious and reject
  else
    Malicious request and reject!
else
  Integrity check failed, reject request!

1) the vehicle is moving from one LI to another within the same CA,
2) the vehicle is moving from one LI to another belonging to two different CAs.

For both cases, let us assume that a vehicle is moving to a new LI region from the current LI, which may or may not be within the same CA. Then, the handover is as follows:

- The vehicle generates a new ephemeral token string β_1 with a new public key pair as $(\mathsf{sk}^*, \mathsf{pk}^*)$ and computes a new pseudoID $\alpha_1 = \mathrm{H}(\mathsf{LIC} \cdot \mathrm{H}_2(\beta_1||\mathsf{t}))$.
- It then generates a pseudoID acquire request $\mathrm{A}_f = \mathrm{Enc}\left(\mathrm{H}(\mathsf{sk}^* \cdot \mathsf{pk}_{\mathcal{C}}), (\beta_1||\alpha_1||\alpha||\mathsf{t})\right)$.
- For the integrity of the request, a signature is generated by the vehicle as $\mathrm{Sign}\left(\mathsf{sk}^*, \mathrm{H}_1(\mathrm{A}_f||\mathrm{H}(\mathsf{pk}^*)||\mathrm{H}(\mathsf{pk}_{\mathcal{C}}))\right) \longrightarrow \sigma_3$
- The vehicle then sends the handover tuple $[\mathrm{A}_f, \mathsf{pk}^*, \mathsf{pk}_{\mathcal{C}}, \sigma_3]$ to the new LI. The tuple consists of the public key $\mathsf{pk}_{\mathcal{C}}$ of the parent CA, where the vehicle is initially registered.
- The LI forwards the tuple to the connected CA. The CA can now check if it is the parent CA for the vehicle; otherwise, simply forward the request to the corresponding parent CA.
- Once received, the parent CA first checks the integrity of the request by verifying the signature as $\mathrm{Ver}\left(pk^*, \mathrm{H}_1(\mathrm{A}_f||\mathrm{H}(\mathsf{pk}^*)||\mathrm{H}(\mathsf{pk}_{\mathcal{C}})), \sigma_3\right) \longrightarrow \{T/F\}$.
- If valid, the parent CA computes $\mathrm{Dec}\left(\mathrm{H}(\mathsf{pk}^* \cdot \mathsf{sk}_{\mathcal{C}}), \mathrm{A}_f\right)$ and extracts the values $\alpha, \alpha_1, \beta_1$ and t.
- If a match with received α is found in the CA's record, it proceeds to the next steps or discards the request.
- Check if the vehicle has been reported by any of its connected LIs previously and currently needs to be revoked. If yes, then discard the request or proceed

to the next step.

- If not revoked, the CA selects the corresponding LIC value from its record and computes $\alpha_1^* = \mathrm{H}\left(\mathsf{LIC} \cdot \mathrm{H}_2\left(\beta_1 || \mathsf{t}\right)\right)$ with received β_1, t, and verify if $\alpha_1^* \stackrel{?}{=} \alpha_1$.
- If matched, the parent CA shares the pair α_1, β_1 with the requesting CA or LI. The vehicle can now perform the ϵ-authentication process with the new LI. If $\alpha_1^* \neq \alpha_1$, the parent CA rejects the request as malicious.
- It is notable that every pseudoID α has an expiry time shared with the requesting LI. Once it expires the LI removes all the α that are no longer valid and the corresponding vehicle performs a handover to update its pseudoID.

V. PRIVACY AND SECURITY ANALYSIS

The well-established elliptic curve cryptography is the security backbone of the proposed scheme and its privacy guarantees. To evaluate the privacy and security capabilities of our proposed scheme, we follow a methodology similar to that of Cui et al.[17], Feng et al.[12], and Inedjaren et al.[14]. We adapt the specific security requirements outlined in section III-B as our security test cases. These requirements are then scrutinized against the designed threat model presented in section III-A, which corresponds with analogous models discussed by Mejri et al.[28] and Zhang et al.[4]. Using this approach, each subprotocol within our proposed scheme undergoes analysis to assess its adherence to the security requirements. We examine secret security parameters, keys, and cryptographic operations against potential adversaries to demonstrate the scheme's security robustness. Furthermore, we conduct a comparative security analysis to clearly illustrate the level of safety assurance offered by our proposed scheme compared to other schemes, as detailed in Table 3.

A. PRIVACY ANALYSIS:

The utmost importance of preserving vehicles' privacy from internal and external adversaries is challenging. Vehicle privacy includes identity-based data, traffic routes, location traceability, and active and idle status. Several privacy challenges are therefore discussed below.

1) Identity Theft

Our scheme achieves resistance against vehicle identity theft by not using original identities or license numbers in V2V or V2I communication. A randomized certificate δ_v is generated using the pseudoID (α) during the V2I authenticated key sharing. The authentication message tuple only allows a semi-trusted LI to relate the tuple to a specific pseudoID α to perform authentication. But the pseudoID is a pseudorandom bitstring providing no information about the vehicle's original identity or user. The original information is only available to a parent CA to which the vehicle is initially registered. Similarly, the signatures σ_1, σ_2 with a public key $\mathsf{pk}_{\mathcal{V}}$ shared during the V2V broadcast do not leak identity-based information. During a handover request, a message tuple from a vehicle $\mathcal{V}$ consists of the pseudoID α, is always encrypted using the randomized symmetric secret key $\mathrm{H}\left(\mathsf{sk}_{\mathcal{V}} \cdot \mathsf{pk}_{\mathcal{C}}\right)$ known only by the parent CA and unknown to any external or internal adversary. Therefore, an internal adversary, such as a curious legitimate vehicle or an external adversary, cannot extract identity-based information from a message tuple sent by a vehicle.

2) Location Traceability and Unlinkability

Location-based traceability of a vehicle $\mathcal{V}$ is blocked from an external adversary using unlinkable authentication parameters sent by $\mathcal{V}$ in each epoch. In our scheme, unlinkability is achieved using pseudorandom values to generate an authentication message tuple in each epoch. Without relating two V2I authentication tuples in any two epochs, ϵ_1 and ϵ_2, other legitimate vehicles in the system cannot trace vehicle $\mathcal{V}$'s traffic route from one epoch to another based on its communication. When vehicle $\mathcal{V}$ moves from one LI region to another, it performs the handover subprotocol with a new public key pair by generating and sharing a new pseudoID α such that, even if the two LIs communicate, they can not relate if two pseudoIDs α and α_1 or public keys $\mathsf{pk}_{\mathcal{V}}, \mathsf{pk}^*$ belong to the exact vehicle $\mathcal{V}$. However, in VANET, vehicles must share critical traffic information with other vehicles throughout an epoch period, allowing them to be traceable only for the epoch duration. As an epoch is short, possibly from a few seconds to a few minutes, the scope of traceability within an epoch is minimal.

3) Active and inactive status

At which time a vehicle $\mathcal{V}$ is active in the network and which vehicles are inactive/inoperative is private information safe from other vehicles in the system and external adversaries. Only the corresponding LI can detect the pseudoID of $\mathcal{V}$ when active as it authenticates the vehicle. When a vehicle does not communicate with the LI, it assumes that either the vehicle has moved to another LI or has become inoperative. LI is a semi-trusted entity, so we presume this status information is safe with the LIs.

B. SECURITY ANALYSIS

Following the security requirements mentioned in Section III, we have analyzed the proposed scheme's security guarantees below.

1) Message confidentiality and intigrety

The content of any message sent during a subprotocol in the scheme is encrypted using the current epoch key (ek) or the one-to-one symmetric key generated for AES encryption. For a V2V broadcast, if the message (m) is encrypted as $\zeta = \mathrm{Enc}\left(\mathsf{ek}, m\right)$ it is only decryptable by the vehicles having the current epoch information and the key ek. This means a non-confidential broadcast is only decryptable by authenticated vehicles in the same LI region at the current epoch and hidden from any external adversary

listening to the communication. Similarly, an external or internal adversary curious to know the content of a shared confidential V2V message will fail to generate the symmetric key $H(\mathsf{sk}_{\mathcal{V}_s} \cdot \mathsf{pk}_{\mathcal{V}_R})$ used to generate cyphertext ζ as $\zeta = \mathrm{Enc}\left(H(\mathsf{sk}_{\mathcal{V}_s} \cdot \mathsf{pk}_{\mathcal{V}_R}), m\right)$; therefore, the confidentiality of the message is always preserved.

Whenever a message is received by a vehicle or an LI, the integrity is verified using the hash function $\mathrm{H}_1()$. For example, when an LI receives an authentication request, the integrity is checked as $\mathrm{H}_1(\alpha||\mathsf{t}||\beta||\mathrm{H}(\mathsf{pk}_{\mathcal{V}})) = \delta_v$. Similarly, an LI acknowledgment is verified by a vehicle as $\mathrm{H}_1(\mathsf{ct}_2||\mathsf{t}||\alpha||\mathrm{H}(\sigma_1)) = \mathsf{ack}$. Using simple hash functions to preserve message integrity, our scheme is secure against message content modification in communication.

2) Mutual verifiability

Any two communicating entities in both V2I and V2V communication must mutually verify their identities before accepting messages. Before sending an authentication/handover tuple to an LI, a vehicle checks the validity of the LI's public key by verifying the corresponding CA's signature on $\mathsf{pk}_{\mathcal{L}}$ as $\mathrm{Ver}(\mathsf{pk}_{\mathcal{C}}, \mathrm{H}(\mathsf{pk}_{\mathcal{L}}), \sigma) \rightarrow \{T, F\}$. Similarly, after receiving a request, LI verifies the vehicle's authenticity before it shares the epoch key and information with the vehicle. Thus, the verifiability is mutual between vehicles and LIs. During V2V direct message sharing, only authenticated vehicles at the current epoch can mutually verify each other's public key signed by the corresponding LI. A receiver vehicle can check the signature σ_1 on the sender vehicle's public key by checking $\mathrm{Ver}\left(\mathsf{pk}_{\mathcal{L}}, \mathrm{H}\left(\mathsf{pk}_{\mathcal{V}_S}\right)||\epsilon, \sigma_1\right) \rightarrow \{T, F\}$ and vice versa. Therefore, the proposed scheme allows only valid entities to verify each other, providing mutual verifiability successfully.

3) Accountability

An authenticated but malicious vehicle in the system can send false messages, making it an internal adversary in the system. In our scheme, whenever a dispute is reported by $\mathcal{V}_R$, the LI immediately enforces the revocation subprotocol to identify the malicious vehicle and put it on the B-list, which will prevent other vehicles from accepting messages from the malicious $\mathcal{V}_S$. During V2V direct message sharing, the receiver $\mathcal{V}_R$ is only allowed to accept a message if the signature σ_1 on the sender's public key $\mathsf{pk}_{\mathcal{V}_S}$ and the signature σ_2 on the message is verified within the current epoch. Therefore, whenever a dispute report is received, the LI confirms that the reporting vehicle $\mathcal{V}_R$ has already verified the disputed $\mathcal{V}_S$'s authenticity. To avoid false dispute reports against any vehicle, the LI verifies the signatures σ_2, σ_3 on the disputed message to confirm if the disputed sender $\mathcal{V}_S$ indeed sent it. However, in our assumption, LI is a semi-trusted entity and, therefore, is not allowed to know the real identity of $\mathcal{V}_S$. If a report is valid, then LI first B-lists $\mathcal{V}_S$ as a malicious vehicle and then reports the incident to $\mathcal{V}_S$'s parent CA, who can disclose the real identity of the malicious vehicle if necessary. If the dispute report is invalid, the LI can hold the reporting vehicle accountable for false reporting. Thus, our proposed scheme achieves privacy-preserving direct accountability in handling disputes.

4) Man-in-the-middle attack

An adversary $\mathcal{A}$ listening to the network can try to intercept messages and perform cryptanalysis to acquire secret and confidential data. This attack is possible each time a vehicle sends any message tuple, such as the V2I authentication and response tuple, the V2V communication tuple, the report tuple, and the handover tuple.

The authentication tuple $[\delta_v, \mathsf{ct}_1, \mathsf{pk}_{\mathcal{V}}, \mathsf{t}]$ consists of a certificate δ_v which is a randomized hash value containing the secret parameters α, β. But as hash functions are irreversible, generating the secret parameters back from δ_v is not possible by $\mathcal{A}$. The ct_1 is an encrypted message with the secret key $\mathrm{H}(\mathsf{sk}_{\mathcal{V}} \cdot \mathsf{pk}_{\mathcal{L}})$, only decryptable by the corresponding LI. The response tuple, $[\mathsf{ack}, \mathsf{ct}_2, \mathsf{t}, \sigma_1]$ also consists of an irreversible hash value (ack) and encrypted ciphertext ct_2. The timestamp t in these tuples is masked into the hashed values; therefore, changing them will be detected by the receiver.

From the V2V communication tuple $[\zeta, \sigma_1, \sigma_2, \mathsf{pk}_{\mathcal{V}_S}, \mathsf{t}]$, the adversary can get a public key and its corresponding signature. But the ciphertext ζ is not decryptable by $\mathcal{A}$ as it does not have the encryption key ek or $\mathrm{H}(\mathsf{sk}_{\mathcal{V}_S} \cdot \mathsf{pk}_{\mathcal{V}_R})$. Also, $\mathcal{A}$ can not check the validity of a public key or a message in transmission as it does not know the epoch information ϵ that is used to sign the public key. Similarly, the report tuple $[\sigma_2, \sigma_3, \zeta, \mathsf{t}, \mathsf{ct}_{\overline{m}}, \mathsf{pk}_{\mathcal{V}_S}, \mathsf{pk}_{\mathcal{V}_R}]$ only allows $\mathcal{A}$ to look into the public keys and signatures, which leak no confidential information. The encrypted messages do not provide any information to an external adversary as it does not have the epoch key (ek) or the encryption key $\mathrm{H}(\mathsf{sk}_{\mathcal{V}_R}.\mathsf{pk}_{\mathcal{L}})$. Even though the adversary can guess which public key might be reported, it does not pose any direct or indirect security threat to the system. The handover tuple $[\mathrm{A}_f, \mathsf{pk}_{\mathcal{V}}]$ and update tuple similarly provides no information to the external or internal adversary.

5) Replay attack

To perform replay attacks, $\mathcal{A}$ first intercepts a message and then sends it again with/without modification to get unauthorized access. The use of timestamps t in the communication tuples protects them from being replayed. As the timestamps are masked with secret parameters into the hashed values or integrated into encrypted values, changing them in the tuple will be easily detectable. For example, the timestamp in the authentication tuple is included in the certificate $\delta_v = \mathrm{H}_2(\alpha||\mathsf{t}||\beta||\mathrm{H}(\mathsf{pk}_{\mathcal{V}}))$; therefore changing them in the tuple is detectable. Also, as α, β are secret parameters, an adversary can not generate δ_v.

A malicious, legitimate vehicle becomes an internal adversary when it tries to reuse a signed public key from one session to send authenticated V2V messages in another

Scheme / Security	Our scheme	Wang et al.[11]	Cui et al.[17]	Yang et al.[8]	Feng et al.[12]	Sikarwar et al.[18]
Privacy Leakage	None	None	Identity	Identity	None	Traceability
Confidentiality/Integrity	Preserved	Preserved	Preserved	Not preserved	Preserved	Not preserved
Mutual Verifiability	Achieved	Achieved	Achieved	Achieved	Not achieved	Achieved
Direct Accountability	Yes	No	No	No	No	No
MITM	Secure	Secure	Secure	Vulnerable	Secure	Vulnerable
Replay Attack	Secure	Vulnerable	Secure	Secure	Secure	Secure
Impersonation/Framing	No	No	No	No	No	Yes
Repudiation	No	Vulnerable	Vulnerable	No	No	Vulnerable
Sybil Attack	Secure	Vulnerable	Vulnerable	Vulnerable	Secure	Vulnerable

TABLE 3: Comparative study on security guarantees provided by proposed scheme vs similar state of the art schemes.

session within the same LI. To avoid this, during V2I key sharing, the LI signs the vehicle's public key with the current epoch information $\mathrm{Sign}\left(\mathsf{sk}_{\mathcal{L}}, \mathrm{H}\left(\mathsf{pk}_{\mathcal{V}}\right)||\epsilon\right) \rightarrow \sigma_1$. Also, during V2V, a receiver vehicle checks the receiver's signature with the current epoch information $\mathrm{Ver}\left(\mathsf{pk}_{\mathcal{L}}, \mathrm{H}\left(\mathsf{pk}_{\mathcal{V}_S}\right)||\epsilon, \sigma_1\right) \rightarrow \{T, F\}$. Therefore, a signature on the public key is only valid for the epoch in which it was generated and signed. Using it in a different epoch will not pass the verification process.

6) Impersonation attack/Framing

The adversary $\mathcal{A}$ can try to impersonate or frame a legitimate vehicle or even an LI to acquire illegal access to the system. To protect the legitimacy of the entities, our scheme imposes an authenticity check solely provided by trusted CAs. Each LI in the system is registered to a particular CA who also signs the LI's public key as $\mathrm{Sign}\left(\mathsf{sk}_{\mathcal{C}}, \mathrm{H}\left(\mathsf{pk}_{\mathcal{L}}\right)\right) \rightarrow \sigma$. To impersonate a valid LI, $\mathcal{A}$ must acquire this signature to pass the public key validity check $\mathrm{Ver}\left(\mathsf{pk}_{\mathcal{C}}, \mathrm{H}\left(\mathsf{pk}_{\mathcal{L}}\right), \sigma\right) \rightarrow \{T, F\}$ by the vehicles. As the CAs decide and verify each LI before installation, impersonating them becomes impossible for $\mathcal{A}$. Similarly, vehicles are initially registered to a parent CA to acquire a secret license number with long-term pseudoID. Therefore, impersonating a vehicle will require $\mathcal{A}$ to access the secret parameters of the vehicle, which are assumed to be safe in the tamper-protected area on the OBU unit of the vehicle.

7) Repudiation

Repudiation takes place when a sender or receiver denies acknowledging a send or receive of a message. In our scheme, the V2V communication and revocation subprotocol could be a potential target for an internal adversary, such as a malicious vehicle, to perform repudiation. To bring non-repudiation in our scheme, a sender vehicle needs to sign a message with its secret key and an epoch-specific value as $\mathrm{Sign}\left(\mathsf{sk}_{\mathcal{V}_S}, \mathrm{H}_1\left(\zeta||\rho\right)\right) \rightarrow (\sigma_2)$. The signature σ_2 and the epoch-specific value $\rho = \mathrm{H}_1\left(\mathsf{ek}||\epsilon||\mathsf{t}\right)$ guarantee the receiver that the sender has signed the message at a specific time t and belongs to the same epoch. A receiver only accepts messages if the sender's public key is valid in the current epoch and if the message is signed with the corresponding secret key by verifying $\mathrm{Ver}\left(\mathsf{pk}_{\mathcal{V}_S}, \mathrm{H}_1\left(\zeta||\rho\right), \sigma_2\right) \rightarrow \{T, F\}$. When a vehicle is reported to the LI, it also checks the signatures on the messages in the same way, to confirm the signature legitimacy of the sender and the receiver at the current epoch. Since the secret key $\mathsf{sk}_{\mathcal{V}}$ of a sender vehicle is kept confidential and used to sign messages, while the corresponding public key $\mathsf{pk}_{\mathcal{V}}$ is the sole key for successful signature verification, this means that a sender cannot deny having signed a message, and a receiver cannot deny receiving a valid signature.

8) Sybil attack

In our scheme, a particular vehicle has only one parent CA to which it is registered. A malicious vehicle trying to register in multiple CAs will be detected in the registration process. Also, a malicious vehicle might try to have multiple instances at any LI by continuously generating and sending handover requests to acquire new pseudoIDs. The CA can easily handle this by analyzing the vehicle's activity, as handover has to be confirmed by the parent CA. If the parent CA detects that the vehicle is requesting different pseudoIDs at the same LI, it can simply reject the request and send a negative acknowledgment to the requesting vehicle.

9) Denial of Service (DoS) attack

A DoS attack aims to make the system resources unavailable to its user. In our case, an external adversary can try to flood the network with fake authentication messages and make the LI busy. However, the step-wise V2I verification process first checks the existence of the ephemeral token β in its database before further processing. This check can be done extremely fast, allowing LI to reject malicious authentication messages in the first step of execution. Similarly, if an already authenticated malicious vehicle as an internal adversary continuously sends multiple authentication requests, the LI can detect its activity while authenticating and reject the malicious requests. Therefore, the step-wise fast verification allows our scheme to deal with flooded malicious requests to be rejected extremely fast, reducing the possibility of DoS attacks significantly.

C. COMPARATIVE SECURITY AND PRIVACY ANALYSIS

Table 3 presents a comparative analysis of security and privacy strength provided by state-of-the-art schemes and

the proposed scheme. The scheme by Wang et al. [11] is secure against traceability and original identity-based privacy thefts and has also achieved resistance against common security attacks such as confidentiality, integrity, MiMA, and framing. However, this scheme is vulnerable to replay attacks as a pseudonym, and its corresponding certificate can be stolen from a V2I authentication request. It can be used by another vehicle or adversary with a new timestamp and location information and can be successfully verified by the RSUs. Also, the scheme is vulnerable to repudiation and sybil attacks, as in V2V message sharing, the receiver vehicle cannot uniquely identify the sender of a message. Therefore, if a malicious sender randomly chooses a framing-free key, the receiver vehicle or CA cannot uniquely identify the sender. Revoking a malicious vehicle will require all the pseudonyms and corresponding certificates of a vehicle to be revoked from all RSUs, making it very inefficient. In the scheme by Cui et al. [17], a vehicle's original identity is also shared with cloud service providers, which can not be fully trusted. Also, the cloud service providers can communicate with each other and trace a vehicle's path easily. It is also vulnerable to repudiation and sybil attacks, and revocation of malicious vehicles is not possible. The authentication request sent by a vehicle to a leader edge node in the scheme proposed byYang et al.[8] includes the original vehicle identity. The request is sent using a public channel; therefore, identity-based privacy confidentiality is not preserved, allowing an adversary to perform MiMA. Even though trust authorities can identify a malicious vehicle, they can not achieve direct accountability by immediately revoking a vehicle for sending malicious messages. In [12], Feng et al. preserves user privacy in communication and protects from most of the common security attacks. However, mutual verifiability is not achieved as a sender always broadcasts non-confidential messages without knowing the receiver. Similar to [8], this scheme is limited to only identifying a malicious vehicle but can not immediately prevent it from sending malicious messages. The scheme by Sikarwar et al. [18] uses pseudonyms similar to the scheme [11], but fails to protect location-based traceability as the public key of a vehicle remains the same and linkable. Also, using the personal ID (Per_ID_2) generated by a vehicle and the public key ($Pubkey_2$), an adversary can generate the private key ($Prikey_2$). Therefore, integrity is lost, allowing adversaries to perform MiMA and replay attacks. Revocation of malicious vehicles is not possible. Heavy computation requirements in the authentication process make all these comparative schemes vulnerable to DoS attacks. The proposed scheme overcomes these issues and guarantees strong security and privacy.

VI. PERFORMANCE ANALYSIS

Scheme efficiency depends on minimizing several parameters, such as computation costs, communication overhead, and overall delay. Computation cost represents the execution delay in performing the cryptographic operations associated with different subprotocols, expressed in milliseconds or ms. Communication overhead is the total size of the message bytes sent during any subprotocol. The overall delay represents the sum of execution delay and transmission latency from the sender to the receiver. Table 5 lists the computation cost for each scheme operation with the number of cryptographic operations performed.

A. SIMULATION SETUP

The simulation of cryptographic operations to assess computation delays in the proposed scheme follows a methodology similar to that used in various VANET schemes introduced by Yang et al. [8], Wei et al. [7], Feng et al. [12], Sikarwar et al.[18] and others. Notably, this simulation strategy focuses solely on the cryptographic functions utilized in the proposed scheme to determine their average execution times. Each cryptographic operation is individually implemented and then executed multiple times to gauge the average computation delay. Importantly, this implementation remains unaffected by parameters such as vehicle speed or status, as these factors do not impact the execution time of cryptographic operations. Rather, the execution time is contingent solely upon the simulation platform and available resources, such as computation power and memory, as previously mentioned.

To extract the execution times of the cryptographic functions used in the protocol, we have used an Intel i7-6500U @ 2.50GHz CPU with two cores and four logical processors having 16GB of physical memory(RAM). We have used a Linux virtual environment and implemented the cryptographic functions in C programming language using the OpenSSL cryptographic library [31]. The security parameter is set to $128-bits$ security; therefore, the AES symmetric key is set to $128-bits$. Similarly, for the ECDSA, we have used the standard cryptographic curve "$secp256k1$", which provides a security level of $128-bits$ (also used in bitcoin technology [32]). The "$secp256k1$"curve is defined on a prime field of size $256-bits$, and the order of the base point P is also $256-bits$.

B. COMPUTATION COSTS

To analyze the computation cost, the average execution time of each cryptographic operation is computed in the scheme and presented in Table 4. During simulation, we assume equal computational capabilities for OBUs, LIs, and CAs, but CAs and LIs typically have greater resources in reality. From Table 5, a vehicle takes $1.6196ms$ to generate an authentication tuple and $1.8317ms$ to get authenticated by an LI. Thus, the total computation cost from sending to acknowledge verification tuple is $(1.6196 + 1.8317 + 0.0054) = 3.4567ms$. For V2V direct messages, from generation to authentication, the scheme requires $(1.0172 + 2.0648) = 3.082ms$ for non-confidential and $(1.8252 + 2.8728) = 4.698ms$ for confidential broadcasts. Generating a report by a vehicle takes $1.015ms$, and LI verification takes $2.8728ms$. Handover request completion

Symbol	Description	Execution time (ms)
$\mathcal{T}_{\mathsf{Sign}}$	The time to execute an ECDSA signature	≈ 1.0113
$\mathcal{T}_{\mathsf{Veri}}$	The time to execute an ECDSA verification	≈ 1.0286
$\mathcal{T}_{\mathsf{pm}}$	The time to execute a point multiplication on curve "$secp256k1$"	≈ 0.8069
$\mathcal{T}_{\mathsf{h}}$	The time to execute a hash operation	≈ 0.0011
$\mathcal{T}_{\mathsf{Enc}}$	The AES encryption time	≈ 0.0026
$\mathcal{T}_{\mathsf{Dec}}$	The AES decryption time	≈ 0.0032

TABLE 4: Execution times of different cryptographic operations in our scheme

Scheme Operations	Cryptographic Executions	Time (ms)
$\mathcal{V}$ generating authentication tuple	$2\mathcal{T}_{\mathsf{pm}} + 3\mathcal{T}_{\mathsf{h}} + \mathcal{T}_{\mathsf{Enc}}$	1.6196
LI verifies the Tuple	$\mathcal{T}_{\mathsf{Dec}} + 7\mathcal{T}_{\mathsf{h}} + \mathcal{T}_{\mathsf{pm}} + \mathcal{T}_{\mathsf{Enc}} + \mathcal{T}_{\mathsf{Sign}}$	1.8317
$\mathcal{V}$ Verifies Acknowledgement	$2\mathcal{T}_{\mathsf{h}} + \mathcal{T}_{\mathsf{Dec}}$	0.0054
$\mathcal{V}$ generates V2V tuple	case 1: $3\mathcal{T}_{\mathsf{h}} + \mathcal{T}_{\mathsf{Enc}} + \mathcal{T}_{\mathsf{Sign}}$ case 2: $4\mathcal{T}_{\mathsf{h}} + \mathcal{T}_{\mathsf{Enc}} + \mathcal{T}_{\mathsf{pm}} + \mathcal{T}_{\mathsf{Sign}}$	1.0172 1.8252
$\mathcal{V}$ verifies V2V tuple	case 1: $4\mathcal{T}_{\mathsf{h}} + 2\mathcal{T}_{\mathsf{Veri}} + \mathcal{T}_{\mathsf{Dec}}$ case 2: $5\mathcal{T}_{\mathsf{h}} + 2\mathcal{T}_{\mathsf{Veri}} + \mathcal{T}_{\mathsf{pm}} + \mathcal{T}_{\mathsf{Dec}}$	2.0648 2.8728
$\mathcal{V}$ generates a report	$\mathcal{T}_{\mathsf{Enc}} + \mathcal{T}_{\mathsf{Sign}} + \mathcal{T}_{\mathsf{h}}$	1.015
LI verifying the report	$5\mathcal{T}_{\mathsf{h}} + 2\mathcal{T}_{\mathsf{Veri}} + \mathcal{T}_{\mathsf{Dec}} + \mathcal{T}_{\mathsf{pm}}$	2.8728
$\mathcal{V}$ generates handover tuple	$6\mathcal{T}_{\mathsf{h}} + \mathcal{T}_{\mathsf{Enc}} + \mathcal{T}_{\mathsf{pm}} + \mathcal{T}_{\mathsf{Sign}}$	1.8274
CA performs a handover	$\mathcal{T}_{\mathsf{Veri}} + \mathcal{T}_{\mathsf{Dec}} + 6\mathcal{T}_{\mathsf{h}} + \mathcal{T}_{\mathsf{pm}}$	1.8453

TABLE 5: Computation cost of all scheme operations in the proposed protocol.

Communication Tuple	Tuple Content	Size (Bytes)
Authentication Tuple	$[\delta_v, \mathsf{ct}_1, \mathsf{pk}_{\mathcal{V}}, \mathsf{t}]$	$(32+48+4+64) = 148$
Acknowledgement Tuple	$[\mathsf{ack}, \mathsf{ct}_2, \mathsf{t}, \sigma_1]$	$(32+56+4+64) = 156$
V2V Tuple (ζ is eliminated)	$[m_h, \sigma_1, \sigma_2, \mathsf{pk}_{\mathcal{V}_S}, \mathsf{t}]$	$(32+64+64+64+4) = 228$
Report Tuple (ζ, $\mathsf{ct}_{\overline{m}}$ are eliminated)	$[m_h, \sigma_2, \sigma_3, \mathsf{t}, \mathsf{pk}_{\mathcal{V}_S}, \mathsf{pk}_{\mathcal{V}_R}]$	$(32+64+64+4+64+64) = 302$
Handover Tuple	$[\mathrm{A}_f, \mathsf{pk}^*, \mathsf{pk}_c a, \sigma_3]$	$(68+64+64+64) = 260$

TABLE 6: Communication overhead or message sizes of each subprotocol in our scheme

requires $(1.8274 + 1.8453) = 3.6727ms$. A comparative analysis is done in Section VI.

C. COMMUNICATION OVERHEAD

With a $128 - bit$ security level, the hash function outputs H, H_1, and H_2 are $16, 32$, and $32 - bytes$ respectively. This makes the secret license LIC, pseudoID α, and ephemeral token β each $16 - bytes$ in size. Utilizing the "$secp256k1$"curve, a private key amounts to $32-bytes$, and a public key occupies $64 - bytes$. For AES encryption in GCM (Galois/Counter Mode) [33], the encrypted ciphertext size equals plaintext size plus an additional $16 - bytes$ for the authentication tag. In V2I authentication, auxiliary information r is fixed at $16 - bytes$.

Table 6 displays the total message sizes for each subprotocol in our scheme, facilitating computation cost analysis. Notably, in V2V message sharing, we've excluded the size of the message tuple, focusing solely on security and authenticity. This omission includes the size of ζ, which represents the encrypted traffic information that vehicles share. Since traffic data varies, we exclude the size of ζ from consideration. Similarly, when determining the report tuple size, we exclude the incident messages ζ and $\mathsf{ct}_{\overline{m}}$ as they fall outside our scope.

D. OVERALL DELAY

Let us consider the average transmission latency of a message from a sender point to a receiver is $\mathcal{L}_{\mathsf{ch}}$. Notably, the transmission latency depends on several parameters, such as the type of communication channel, the topology, vehicle speed, etc. However, for simplicity in the simulated case study and consistency in comparative studies, we have considered a generalized representation of these factors by $\mathcal{L}_{\mathsf{ch}}$. To analyze the overall delay, we combine computation costs as execution delays and then add transmission latency.

Overall Delay = Total Execution Delay + Total Transmission Latency

In the proposed scheme, a vehicle's total execution delay for V2I authenticated key sharing is $3.4567ms$, requiring two transmissions (vehicle to LI, LI to vehicle). Thus, the overall latency for V2I communication is $(3.4567+2*\mathcal{L}_{\mathsf{ch}})ms$. V2V non-confidential execution delay is $3.082ms$, with a single transmission directly from vehicle to vehicle, resulting in an overall V2V delay of $(3.082 + \mathcal{L}_{\mathsf{ch}})ms$. A confidential direct V2V broadcast requires $(4.6980 + \mathcal{L}_{\mathsf{ch}})ms$. A report takes an overall delay of $(1.015 + 2.8728 + \mathcal{L}_{\mathsf{ch}})ms = (3.8878 + \mathcal{L}_{\mathsf{ch}})ms$, while a handover from vehicle to CA entails an overall delay of $(3.6727+2\mathcal{L}_{\mathsf{ch}})ms$. For efficiency comparison, see Table 7, which represents delay reduction with similar state-of-the-art schemes.

VII. COMPARATIVE STUDIES

In this section, we have analyzed and discussed the efficiency gains and advantages of our proposed distributed authentication scheme in decentralized VANETs compared to similar state-of-the-art schemes.

A. COMPARATIVE EFFICIENCY ANALYSIS

To evaluate the efficiency of our protocol in reducing computation cost, communication overhead, and overall delay, we conducted a comparative analysis by benchmarking it against state-of-the-art protocols from existing literature. For a fair comparison, we have chosen only those schemes that have also proposed distributed authentication techniques for VANETs using ECC-based, bi-linear-based, and PKI-based protocols. From Table 1, we specifically selected two ECC-based schemes by Cui et al.[17] and Wei et al.[7], which provide an authenticated V2I key-sharing method. We also considered two bilinear-pairing-based schemes by Feng et al.[12] and by Sikarwar et al.[18], chosen for their exclusive authenticated direct V2V message-sharing capabilities. In addition, we included two more schemes by Wang et al.[11] and Yang et al.[8], which, like our own protocol, offer both V2I authenticated key/token sharing and direct V2V message sharing.

For this comparative analysis, we implemented these selected protocols under the same simulation conditions and measured the execution time for the cryptographic operations they entail. The simulation revealed that a bilinear pairing operation takes approximately $\mathcal{T}_{\mathsf{bp}} \approx 2.0132ms$, while an exponentiation operation with two 128-bit values requires $\mathcal{T}_{\mathsf{exp}} \approx 0.0253ms$. The execution time for the Lagrange interpolation operation is approximately $\mathcal{T}_{\mathsf{lag}} \approx 0.1725ms$.

Table 7 presents a comparative efficiency analysis to perform an authenticated V2I key sharing of our protocol and the selected protocols, all achieving a security level of $128-bits$. We considered the total execution and communication requirements for a vehicle to perform authentication in each protocol. The total latencies are considered by the number of communications needed by different entities in the system. For example, $6\mathcal{L}_{\mathsf{ch}}$ for scheme [17] means that the protocol requires six message transfers between various entities in the system to complete a V2I authentication and key sharing. For all the schemes assuming that the underlying channel has an ultra-low latency of $\mathcal{L}_{\mathsf{ch}} \approx 1ms$ as mentioned in [34], [35], we evaluated the overall delay by computing the sum of total execution delay and total transmission latency. To evaluate the efficiency gain in delay reduction, we incorporate a multiplicative coefficient by using the following formula:

$$\textbf{Multiplicative Efficiency Coefficient} = \frac{\textbf{Delay of B}}{\textbf{Delay of A}}$$

where the multiplicative efficiency coefficient represents if the comparative scheme (B) is faster (< 1) or slower (> 1) than our scheme (A). In other words, the multiplicative efficiency coefficient shows if our proposed scheme is relatively fast or slow compared to other schemes. If the value of the multiplicative co-efficient is < 1, it means that the comparative scheme has a lower delay and is faster than our scheme. In contrast, if it is > 1, our scheme has a lower delay and is faster than the comparative scheme. The exact computed value of the multiplicative efficiency coefficient measures how much our proposed scheme is faster or slower than other schemes in terms of delay. Analysis of Table 7 reveals that compared to our proposed scheme, other ECC-based approaches incur a longer delay that makes them, at most, around 2.3 times slower when executing V2I authenticated key sharing. Moreover, pairing-based protocols exhibit a much longer delay that can be as high as ≈ 3.9 times our proposed scheme's delay, making them ≈ 3.9 times slower in V2I authenticated key sharing.

Similarly, Table 8 presents a comparative efficiency to perform directly authenticated V2V message sharing of our protocol and the selected protocols. The V2V execution delay represents the total execution time required from generating a message by the sender to verifying it at the receiver. The required channel latency is $\mathcal{L}_{\mathsf{ch}}$ for all the schemes as we have only selected protocols that achieve direct V2V message sharing. The comparative efficiency gain From Table 8 shows that in comparison with our proposed scheme, the comparative state-of-the-art schemes have a maximum of ≈ 7.5 times longer delay, making them ≈ 7.5 times slower in performing V2V authenticated message sharing. Note that we have considered the computation time for non-confidential broadcast in V2V message sharing for comparative fairness as none of the other selected comparative schemes provides a V2V confidential message sharing.

An alert within V2V communication signifies a message containing time-sensitive information necessitating immediate transmission and verification. As mentioned earlier, transmitting emergency alerts with proximity considerations aimed at collision avoidance is critical for V2V communication. Particularly for vehicles operating at high speeds, it becomes imperative to generate and send alert messages swiftly, ensuring prompt delivery of the message to the receiver(s). In scenarios where protocols involve substantial computation durations for transmitting and receiving alert messages, there's a risk that by the time the receiving vehicle(s) get the alert, they may have already traveled far enough that avoiding a collision becomes impossible. Therefore, reducing V2V execution time in generating and verifying messages directly facilitates vehicles in executing and transmitting emergency messages quickly, even when traveling at high speeds and in close proximity to other vehicles. Considering $\mathcal{L}_{\mathsf{ch}} \approx 1ms$ [34], our scheme requires $\approx (3.3753 + 1) = 4.3753ms$ to successfully send and receive an alert using the same hardware simulation setup. This is significantly lower than the time required to execute and send a V2V message in other comparative schemes presented in Table 8. With the use of lightweight cryptographic operations and keeping the number of operations low in sending and receiving direct V2V authenticated

Schemes	Execution Delay: Key Sharing	Total Transmission Latency	Overall V2I Delay (ms)	Delay Comparison to our scheme	Communication Cost ($bytes$)
Our Scheme	$3\mathcal{T}_{\mathsf{pm}}+2\mathcal{T}_{\mathsf{Enc}}+2\mathcal{T}_{\mathsf{Dec}}+\mathcal{T}_{\mathsf{Sign}}+12\mathcal{T}_{\mathsf{h}}$	$2\mathcal{L}_{\mathsf{ch}}$	$\approx 5.4567\ ms$	(Self comparison)	304
Yang et al. [8]	$7\mathcal{T}_{\mathsf{bp}}+6\mathcal{T}_{\mathsf{pm}}+8\mathcal{T}_{\mathsf{h}}$	$2\mathcal{L}_{\mathsf{ch}}$	$\approx 21.362\ ms$	≈ 3.9 times slower	336
Cui et al. [17]	$8\mathcal{T}_{\mathsf{pm}}+25\mathcal{T}_{\mathsf{h}}$	$6\mathcal{L}_{\mathsf{ch}}$	$\approx 12.4827\ ms$	≈ 2.3 times slower	656
Wang et al. [11]	$2\mathcal{T}_{\mathsf{bp}}+5\mathcal{T}_{\mathsf{exp}}+2\mathcal{T}_{\mathsf{Sign}}+3\mathcal{T}_{\mathsf{Veri}}+\mathcal{T}_{\mathsf{Enc}}+\mathcal{T}_{\mathsf{Dec}}$	$2\mathcal{L}_{\mathsf{ch}}$	$\approx 11.2671\ ms$	≈ 2 times slower	420
Wei et al. [7]	$6\mathcal{T}_{\mathsf{pm}}+6\mathcal{T}_{\mathsf{lag}}+38\mathcal{T}_{\mathsf{h}}$	$4\mathcal{L}_{\mathsf{ch}}$	$\approx 9.9182\ ms$	≈ 1.8 times slower	596

TABLE 7: In above we present a comparative efficiency analysis of V2I authenticated key sharing with state-of-the-art schemes and our proposed scheme. The execution delay reports the amount of delays obtained from the cryptographic operations. The transmission latency amounts the total channel communication delays. The overall delay provides an estimated delay in ms by summing up the execution delays, with the total transmission latency assuming $\mathcal{L}_{\mathsf{ch}} \approx 1\ ms$. The delay comparison provides a multiplicative coefficient representing if the comparative scheme is slower(> 1) or faster (< 1) than our proposed scheme.

Schemes	Execution Delay: Message Sharing	Total Transmission Latency	Overall V2V Delay (ms)	Delay Comparison to our scheme	Communication Cost ($bytes$)
Our Scheme	$\mathcal{T}_{\mathsf{Enc}}+\mathcal{T}_{\mathsf{Dec}}+\mathcal{T}_{\mathsf{Sign}}+2\mathcal{T}_{\mathsf{Veri}}+7\mathcal{T}_{\mathsf{h}}$	$\mathcal{L}_{\mathsf{ch}}$	$\approx 4.082\ ms$	(Self comparison)	228
Feng et al. [12]	$6\mathcal{T}_{\mathsf{bp}}+21\mathcal{T}_{\mathsf{pm}}+22\mathcal{T}_{\mathsf{exp}}\ ms$	$\mathcal{L}_{\mathsf{ch}}$	$\approx 30.5807\ ms$	≈ 7.5 times slower	772
Sikarwar et al.[18]	$3\mathcal{T}_{\mathsf{bp}}+7\mathcal{T}_{\mathsf{pm}}+5\mathcal{T}_{\mathsf{h}}$	$\mathcal{L}_{\mathsf{ch}}$	$\approx 12.8734\ ms$	≈ 3.1 times slower	192
Yang et al. [8]	$3\mathcal{T}_{\mathsf{bp}}+3\mathcal{T}_{\mathsf{pm}}+4\mathcal{T}_{\mathsf{h}}$	$\mathcal{L}_{\mathsf{ch}}$	$\approx 9.4647\ ms$	≈ 2.3 times slower	260
Wang et al. [11]	$3\mathcal{T}_{\mathsf{bp}}+3\mathcal{T}_{\mathsf{exp}}+3\mathcal{T}_{\mathsf{h}}$	$\mathcal{L}_{\mathsf{ch}}$	$\approx 7.1188\ ms$	≈ 1.7 times slower	260

TABLE 8: In above we present comparative efficiency analysis of V2V authenticated key sharing with state-of-the-art schemes and our proposed scheme. The execution delay reports the amount of delays obtained from the cryptographic operations. The transmission latency amounts the total channel communication delays. The overall delay provides an estimated delay in ms by summing up the execution delays, with the total transmission latency assuming $\mathcal{L}_{\mathsf{ch}} \approx 1\ ms$. The delay comparison provides a multiplicative coefficient representing if the comparative scheme is slower(> 1) or faster (< 1) than our proposed scheme.

messages, we have achieved a high-efficiency gain in V2V alert sharing.

B. DISCUSSIONS

Below, we summarize the insights of the comparative studies.

1) The proposed distributed VANET architecture offers significant advantages over traditional centralized VANETs, particularly in terms of scalability, fault tolerance, and overall reliability. By distributing tasks among multiple CAs, the architecture eliminates bottlenecks and enables efficient task distribution as the network scales. Additionally, the architecture ensures fault tolerance by tolerating failures of both CAs and LIs, with adaptive control mechanisms to manage network disruptions. Moreover, the distributed nature of the architecture enhances security and privacy through collaborative security defenses and improves resource availability by eliminating single points of failure. These features collectively contribute to a more robust and reliable VANET communication system, capable of meeting the demands of diverse and dynamic environments.
2) The proposed distributed LI-based authentication scheme effectively addresses scalability limitations and reliability issues inherent in centralized authentication mechanisms in high-density VANETs. By decentralizing authentication and assigning Local Inspectors (LIs) to verify requests locally, the system mitigates concerns such as CA overloading, network congestion, and reliance on a single point of failure. With LIs handling authentication regionally, the impact of hardware or communication failures in specific areas on overall system functionality is minimized, ensuring high reliability. Furthermore, by bypassing the need for authentication requests to reach centralized authorities, our scheme eliminates additional delays associated with CA operations, promoting faster response times and enhancing reliability, particularly in time-sensitive communication scenarios within dense VANET environments.
3) The immediate revocation feature in the VANET environment significantly enhances security by swiftly blacklisting the public key of disputed senders upon verification of a valid report. This proactive measure, as detailed in the protocol, enables Local Inspectors (LIs) to broadcast the blacklist to their respective regions promptly, ensuring that all valid vehicles cease

receiving malicious messages from the identified malicious senders. By preventing the propagation of these harmful messages, the immediate revocation feature effectively mitigates the risks associated with compromised communication channels, thereby bolstering safety for drivers and passengers within the VANET ecosystem.

4) The detailed privacy and security analysis in Section V, and the comparative analysis in Table 3 make it clear that our scheme is secure and can protect against various security attacks to which similar schemes are vulnerable. Therefore, our proposed scheme provides better passenger safety in VANET communications.
5) The performance analysis in Section VI and the comparative efficiency analysis from Table 7 and Table 8 show that our scheme requires comparatively less computation cost and overall delay in both V2V key sharing and V2V message sharing. The efficiency of the proposed V2I key sharing and V2V message sharing in VANETs heavily relies on employing lightweight cryptographic techniques such as symmetric key encryption/decryption and simple hash functions. By minimizing ECC-based operations, limiting protocol-specific variables, utilizing non-interactive communication, and reducing third-party involvement, the scheme significantly decreases computation and communication costs while maintaining a high level of security. Additionally, the shorter travel path of authentication requests in distributed authentication further enhances efficiency by eliminating delays associated with centralized VANET architectures.
6) Even though the simulation results are platform-dependent, meaning that the execution times for cryptographic operations highly depend on the selected cryptographic library, simulation software, and hardware capabilities, our scheme will consistently outperform the selected state-of-the-art comparative schemes regarding computation, communication, and overall delay due to the selection of lightweight cryptographic operations and reduction in trust party involvements.

VIII. CONCLUSIONS

This research addresses critical privacy and security challenges in Vehicular Ad Hoc Networks (VANETs) by introducing a novel hierarchical decentralized VANET authentication system. The proposed system eliminates single-point failures, enhances user autonomy during registration, and enables efficient and privacy-preserving Vehicle-to-Infrastructure (V2I) and Vehicle-to-Vehicle (V2V) communication. It offers swift and privacy-preserving vehicle revocation and supports close proximity alerts, allowing quick emergency responses. These contributions represent a significant step towards mitigating VANET security concerns, reducing latency, and prioritizing the privacy and security of vehicles in V2V and V2I communication. By eliminating the communication bottlenecks and efficiently offering immediate revocation of malicious vehicles, the proposed solution offers a promising advancement in the field of VANET connectivity and security. This research paves the way for a more resilient and responsive VANET ecosystem that can significantly enhance road safety and traffic management while safeguarding sensitive data and ensuring prompt communication in critical situations.

REFERENCES

[1] M. S. Sheikh and J. Liang, "A Comprehensive Survey on VANET Security Services in Traffic Management System," Wireless Communications and Mobile Computing, vol. 2019, pp. 1–23, 2019.
[2] F. Azam, S. K. Yadav, N. Priyadarshi, S. Padmanaban, and R. C. Bansal, "A Comprehensive Review of Authentication Schemes in Vehicular Ad-Hoc Network," IEEE access, vol. 9, pp. 31 309–31 321, 2021.
[3] R. Hussain, J. Lee, and S. Zeadally, "Trust in VANET: A Survey of Current Solutions and Future Research Opportunities," IEEE Transactions on intelligent transportation systems, vol. 22, no. 5, pp. 2553–2571, 2020.
[4] J. Zhang, Q. Zhang, X. Lu, and Y. Gan, "A Novel Privacy-Preserving Authentication Protocol Using Bilinear Pairings for the VANET Environment," Wireless Communications and Mobile Computing, vol. 2021, pp. 1–13, 2021.
[5] K. Gu, K. Wang, X. Li, and W. Jia, "Multi-Fogs-based Traceable Privacy-Preserving Scheme for Vehicular Identity in Internet of Vehicles," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 8, pp. 12 544–12 561, 2021.
[6] C. Yang, P. Jiang, and L. Zhu, "Accelerating Decentralized and Partial-Privacy Data Access for VANET via Online/Offline Functional Encryption," IEEE Transactions on Vehicular Technology, vol. 71, no. 8, pp. 8944–8954, 2022.
[7] L. Wei, J. Cui, H. Zhong, I. Bolodurina, and L. Liu, "A Lightweight and Conditional Privacy-Preserving Authenticated Key Agreement Scheme with Multi-TA Model for Fog-based VANETs," IEEE Transactions on Dependable and Secure Computing, 2021.
[8] A. Yang, J. Weng, K. Yang, C. Huang, and X. Shen, "Delegating Authentication to Edge: A Decentralized Authentication Architecture for Vehicular Networks," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 2, pp. 1284–1298, 2020.
[9] T. Chatterjee, R. Karmakar, G. Kaddoum, S. Chattopadhyay, and S. Chakraborty, "A Survey of VANET/V2X Routing from the Perspective of Non-Learning-and Learning-based Approaches," IEEE Access, vol. 10, pp. 23 022–23 050, 2022.
[10] L. Zhang, Q. Wu, J. Domingo-Ferrer, B. Qin, and C. Hu, "Distributed Aggregate Privacy-Preserving Authentication in VANETs," IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 3, pp. 516–526, 2016.

[11] P. Wang and L. Yining, "SEMA: Secure and Efficient Message Authentication Protocol for VANETs," IEEE systems journal, vol. 15, no. 1, pp. 846–855, 2021.
[12] X. Feng, Q. Shi, Q. Xie, and L. Wang, "P2BA: A Privacy-Preserving Protocol with Batch Authentication Against Semi-Trusted RSUs in Vehicular Ad Hoc Networks," IEEE Transactions on Information Forensics and Security, vol. 16, pp. 3888–3899, 2021.
[13] X. Li, T. Jing, R. Li, H. Li, X. Wang, and D. Shen, "BDRA: Blockchain and Decentralized Identifiers Assisted Secure Registration and Authentication for VANETs," IEEE Internet of Things Journal, 2022.
[14] Y. Inedjaren, M. Maachaoui, B. Zeddini, and J.-P. Barbot, "Blockchain-based Distributed Management System for Trust in VANET," Vehicular Communications, vol. 30, p. 100350, 2021.
[15] M. Saad, M. B. Ahmad, M. Asif, M. K. Khan, T. Mahmood, and M. T. Mahmood, "Blockchain-Enabled VANET for Smart Solid Waste Management," IEEE Access, vol. 11, pp. 5679–5700, 2023.
[16] M. Amara and A. Siad, "Elliptic Curve Cryptography and Its Applications," in International Workshop on Systems, Signal Processing and Their Applications, WOSSPA. IEEE, 2011, pp. 247–250.
[17] J. Cui, X. Zhang, H. Zhong, J. Zhang, and L. Liu, "Extensible Conditional Privacy Protection Authentication Scheme for Secure Vehicular Networks in a Multi-Cloud Environment," IEEE Transactions on Information Forensics and Security, vol. 15, pp. 1654–1667, 2019.
[18] D. D. Sikarwar, Himani, "Towards Lightweight Authentication and Batch Verification Scheme in IoV," IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 5, pp. 3244–3256, 2021.
[19] A. Waheed, M. A. Shah, A. Khan, C. Maple, and I. Ullah, "Hybrid Task Coordination Using Multihop Communication in Volunteer Computing-based VANETs," Sensors, vol. 21, no. 8, p. 2718, 2021.
[20] Z. H. Ali, M. M. Badawy, and H. A. Ali, "A Novel Geographically Distributed Architecture based on Fog Technology for Improving Vehicular Ad hoc Network (VANET) Performance," Peer-to-Peer Networking and Applications, vol. 13, no. 5, pp. 1539–1566, 2020.
[21] F. A. Ghaleb, F. Saeed, M. Al-Sarem, B. Ali Saleh Alrimy, W. Boulila, A. Eljialy, K. Aloufi, and M. Alazab, "Misbehavior-Aware on-Demand Collaborative Intrusion Detection System Using Distributed Ensemble Learning for VANET," Electronics, vol. 9, no. 9, p. 1411, 2020.
[22] Z. Ma, J. Zhang, Y. Guo, Y. Liu, X. Liu, and W. He, "An Efficient Decentralized Key Management Mechanism for VANET with Blockchain," IEEE Transactions on Vehicular Technology, vol. 69, no. 6, pp. 5836–5849, 2020.
[23] M. Arif, G. Wang, M. Z. A. Bhuiyan, T. Wang, and J. Chen, "A Survey on Security Attacks in VANETs: Communication, Applications and Challenges," Vehicular Communications, vol. 19, p. 100179, 2019.
[24] G. Nardini, D. Sabella, G. Stea, P. Thakkar, and A. Virdis, "Simu5G–An OMNeT++ Library for End-to-End Performance Evaluation of 5G Networks," IEEE Access, vol. 8, pp. 181 176–181 191, 2020.
[25] X. Wang, S. Mao, and M. X. Gong, "An Overview of 3GPP Cellular Vehicle-to-Everything Standards," GetMobile: Mobile Computing and Communications, vol. 21, no. 3, pp. 19–25, 2017.
[26] V. Jindal and P. Bedi, "Vehicular Ad-Hoc Networks: Introduction, Standards, Routing Protocols and Challenges," International Journal of Computer Science Issues (IJCSI), vol. 13, no. 2, p. 44, 2016.
[27] C. R. Storck and F. Duarte-Figueiredo, "A Survey of 5G Technology Evolution, Standards, and Infrastructure Associated With Vehicle-to-Everything Communications by Internet of Vehicles," IEEE Access, vol. 8, pp. 117 593–117 614, 2020.
[28] M. N. Mejri, J. Ben-Othman, and M. Hamdi, "Survey on VANET Security Challenges and Possible Cryptographic Solutions," Vehicular Communications, vol. 1, no. 2, pp. 53–66, 2014.
[29] D. Dolev and A. Yao, "On The Security of Public Key Protocols," IEEE Transactions on Information Theory, vol. 29, no. 2, pp. 198–208, 1983.
[30] D. R. Brown, "Generic Groups, Collision Resistance, and ECDSA," Designs, Codes and Cryptography, vol. 35, no. 1, pp. 119–152, 2005.
[31] E. Käsper, "Fast Elliptic Curve Cryptography in OpenSSL," in Financial Cryptography and Data Security: FC 2011 Workshops, RLCPS and WECSR 2011, Rodney Bay, St. Lucia, February 28-March 4, 2011, Revised Selected Papers 15. Springer, 2012, pp. 27–39.
[32] J. W. Bos, J. A. Halderman, N. Heninger, J. Moore, M. Naehrig, and E. Wustrow, "Elliptic Curve Cryptography in Practice," in Financial Cryptography and Data Security: 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers 18. Springer, 2014, pp. 157–175.
[33] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "Efficient and High-Performance Parallel Hardware Architectures for the AES-GCM," IEEE Transactions on Computers, vol. 61, no. 8, pp. 1165–1178, 2011.
[34] R. Hussain, F. Hussain, and S. Zeadally, "Integration of VANET and 5G Security: A Review of Design and Implementation Issues," Future Generation Computer Systems, vol. 101, pp. 843–864, 2019.
[35] M. A. Ferrag, L. Maglaras, A. Argyriou, D. Kosmanos, and H. Janicke, "Security for 4G and 5G Cellular Networks: A Survey of Existing Authentication and Privacy-Preserving Schemes," Journal of Network and Computer Applications, vol. 101, pp. 55–82, 2018.

SUJASH NASKAR (Member, IEEE) received a B.Sc. degree in computer science in 2017 and an M.Sc. degree in 2019 from University of Calcutta, Kolkata, India. He is currently pursuing a Ph.D. degree in IoT Security and Privacy at STC research group, Mid Sweden University, Sundsvall, Sweden.

From November 2022 to July 2023, he was a guest research fellow at Simula UiB, Bergen, Norway. His research interests include privacy and security management for Vehicular ad hoc Networks (VANET), especially developing lightweight privacy-preserving authentication schemes for VANET.

Mr. Sujash's awards and honors include receiving the prestigious Ericsson Grant for a long-term research visit.

CARLO BRUNETTA received his BSc and MSc in mathematics from University of Trento, Trento, Italy, in 2014 and 2016 respectively. He obtained a PhD in cryptology and privacy-preserving protocols from Chalmers University of Technology, Gothenburg, Sweden, in 2021. From 2021 to 2023, he was a researcher in theoretical cryptography and side-channel attacks in cryptography-oriented applications at Simula UiB, Bergen, Norway. He is currently an independent researcher in several aspects of cryptology, applied and not.

GERHARD HANCKE (M'1999,SM'2011,F'2022) is a Professor in the Department of Computer Science at City University of Hong Kong. He received B.Eng and M.Eng degrees in Computer Engineering from the University of Pretoria, South Africa, in 2002 and 2003, and a PhD in Computer Science from the University of Cambridge, United Kingdom, in 2009. Previously he worked as a researcher with the Smart Card and IoT Security Centre and as teaching fellow with the Department of Information Security, both at Royal Holloway, University of London. His research interests are in security, reliable communication and distributed sensing for the Industrial Internet-of-Things.

TINGTING ZHANG received the B.Sc. degree and M.Sc. degree in computer science and engineering from Fudan University, Shanghai, China, in 1982 and 1984, respectively, and the Ph.D. degree in computer science and engineering from Linköping University, Linköping, Sweden, in 1993. She is a Professor of computer engineering with the Department of Information and Communication Systems, Mid Sweden University, Sundsvall, Sweden. Her current research work is in the areas of wireless sensor networks and security.

MIKAEL GIDLUND (M'98-SM'16) received the Licentiate of Engineering degree in radio communication systems from the KTH Royal Institute of Technology, Stockholm, Sweden, in 2004, and the Ph.D. degree in electrical engineering from Mid Sweden University, Sundsvall, Sweden, in 2005. From 2008 to 2015, he was a Senior Principal Scientist and Global Research Manager of Wireless Technologies with ABB Corporate Research, Västerås, Sweden. From 2007 to 2008, he was a Project Manager and a Senior Specialist with Nera Networks AS, Bergen, Norway. From 2006 to 2007, he was a Research Engineer and a Project Manager with Acreo AB, Hudiksvall, Sweden. Since 2015, he has been a Professor of Computer Engineering at Mid Sweden University. He holds more than 20 patents (granted and pending) in the area of wireless communication. His current research interests include wireless communication and networks, wireless sensor networks, access protocols, and security. Dr. Gidlund is an Associate Editor of the *IEEE Transactions on Industrial Informatics.* and *IEEE Journal of Emerging and Selected Topics in Industrial Electronics*