Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.055184

ARTICLE

Trust Score-Based Malicious Vehicle Detection Scheme in Vehicular Network Environments

Wenming Wang[1,2,3,*], Zhiquan Liu[1], Shumin Zhang[1] and Guijiang Liu[1]

[1]School of Computer and Information, Anqing Normal University, Anqing, 246133, China

[2]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, 210023, China

[3]Yunnan Key Laboratory of Service Computing, Yunnan University of Finance and Economics, Kunming, 650221, China

*Corresponding Author: Wenming Wang. Email: wy523@aqnu.edu.cn

Received: 19 June 2024 Accepted: 13 September 2024 Published: 18 November 2024

ABSTRACT

Advancements in the vehicular network technology enable real-time interconnection, data sharing, and intelligent cooperative driving among vehicles. However, malicious vehicles providing illegal and incorrect information can compromise the interests of vehicle users. Trust mechanisms serve as an effective solution to this issue. In recent years, many researchers have incorporated blockchain technology to manage and incentivize vehicle nodes, incurring significant overhead and storage requirements due to the frequent ingress and egress of vehicles within the area. In this paper, we propose a distributed vehicular network scheme based on trust scores. Specifically, the designed architecture partitions multiple vehicle regions into clusters. Then, cloud supervision systems (CSSs) verify the accuracy of the information transmitted by vehicles. Additionally, the trust scores for vehicles are calculated to reward or penalize them based on the trust evaluation model. Our proposed scheme demonstrates good scalability and effectively addresses the main cause of malicious information distribution among vehicles. Both theoretical and experimental analysis show that our scheme outperforms the compared schemes.

KEYWORDS

Distributed; trust mechanism; vehicular network; privacy protection

1 Introduction

Vehicular network represents a fusion of real-time sensing, communication, and computation that transcends their role as mere transportation. From a consumer's perspective, they resemble scintillating electronic to in the field of consumer electronics. As a novel paradigm of network application, it has emerged as a focal point of widespread attention and research. Particularly, the advent of 5G technology has ushered in significant transformations and developmental opportunities in this area [1]. By leveraging advanced on-board units equipped with superior sensing, communication, and networking capabilities, vehicles can establish communication links with neighboring vehicles or infrastructures. This enables the widespread dissemination of traffic information [2]. Not only does this convergence of in-vehicle networks improve driving safety and efficiency, but it also revolutionizes urban transportation systems through intelligent traffic management.

Despite offering significant convenience, vehicular networking presents numerous challenges, particularly in security and privacy protection. First, vehicles face difficulties in quickly verifying the accuracy of the shared information. In the event that malicious vehicles deliberately transmit misinformation, the implications may be severe, even endangering drivers' life. Consequently, it is critical for vehicles to develop a mechanism to monitor and control malicious activities of vehicles [3]. Simultaneously, trust issues arise in communication and collaboration between vehicles. Thus, it is imperative to establish a reliable trust system among devices to prevent malicious device attacks [4]. Additionally, vehicular networks must meet requirements for real-time response, reliability, and scalability to navigate the complex and dynamic traffic conditions [5].

In order to actualize the detection of malicious vehicles, researchers have proposed solutions involving algorithmic models, cryptographic techniques, and identity verification. For example, by dynamically collaborating with the vehicle and the surrounding roadside units, and employing sequential detection algorithms, the accuracy of malicious vehicle identification is enhanced [6]. Building on this, some scholars introduced the Evolutionary Public Goods Game (EPGG) model to detect malicious information. Additionally, a countermeasure against such malicious information is implemented through broadcasting anti-malicious rumor information [7]. To address the potential cross-domain issues in the transmission and sharing of vehicular network data across different regions, a federated detection hierarchical mechanism is proposed. This mechanism facilitates a unified evaluation among devices, expedites identity authentication, and increases the detection accuracy of malicious devices [8]. However, the blockchain in this scheme, aimed at constructing a trust platform, suffers from low detection efficiency and limited scalability. Consequently, it is unsuitable for vehicular network scenarios that require high real-time performance.

In response to the aforementioned issues, this paper proposes an effective trust evaluation mechanism for distributed vehicular networks. This mechanism is intended to counteract the malicious behavior of vehicles spreading misleading messages. Our main contributions are summarized as follows:

(1) We employ a trust aggregation approach, coupled with a trust evaluation model, to calculate initial trust scores for vehicles. Subsequently, we thoroughly analyze vehicle data to refine the trust scores of each vehicle, thereby assessing their credibility. Finally, we introduce cloud supervision systems (CSSs) to verify the authenticity of the information transmitted by vehicles. Malicious vehicles spreading misleading information will be punished, whereas honest vehicles will gain bonus points, thus enhancing the system's environmental credibility.

(2) To enhance the robustness of the trust mechanism in vehicular networks, we propose a distributed architecture comprising the trusted authority institution (TAI) and CSSs. Among them, TAI is the core node which manages and coordinates the trust evaluation process, CSSs manages system-wide trust evaluation, process addressing various scenarios and requirements. This setup promotes labor division, cooperation, and performance optimization.

(3) To efficiently utilize the resource characteristics of CSSs, we introduce a heartbeat mechanism for their scheduling and load balancing. Through this mechanism, CSSs are allowed to periodically transmit their availability information to TAI, which can then make scheduling decisions for each CSS's resources. This achieves optimal load balancing and enhances system performance.

The rest of this paper is organized as follows. Section 2 introduces the related work. The system and attack model are presented in Section 3. Section 4 details the specific scheme. A safety proof and analysis are performed in Section 5. An overview of performance evaluation is provided in Section 6.

Section 7 discusses the feasibility and limitations of the proposed scheme. Finally, Section 8 concludes the paper.

2 Related Work

A. Applications of Trust Management Mechanism

Faced with a growing number of vehicles and diverse service demands, both academia and business are striving to promulgate novel paradigms to enhance road safety by transmitting traffic information via vehicles. Concurrently, a trust management mechanism has emerged as a key solution to prevent malevolent vehicles from spreading false information and harming the system. This mechanism could include three trust models: a Context and Entity-oriented Trust Model [9,10], a Data-oriented Trust Model [11], and a Combined Trust Model [12]. Furthermore, deploying a reputation management mechanism can help establish trust relationships amongst vehicles [13]. Based on this concept, Tian et al. [14] devised a reputation-based architecture, wherein vehicles are necessitated to invest money in reporting accurate traffic events to gain reputation scores. However, this scheme has been identified to have two deficiencies. First, the centralization poses a significant challenge [15]. Without trustworthy Roadside Units (RSUs), the reliance solely on a single central server for traffic information verification risks system paralysis from potential attacks. Trust aggregation through different network paths can help to consolidate a single trust value, enhancing reliability [16]. Therefore, designing a more secure and reliable communication mechanism is crucial to ensure the system's overall reliability and security. Simultaneously, it is necessary to ascertain how to establish a trustworthy interaction mechanism among vehicles to mitigate malicious behavior more effectively. In summary, trust management mechanisms are pivotal for maintaining the integrity and reliability of vehicular networks.

B. Applications of Blockchain Technology

Considering the centralization dilemmas mentioned above, many academics have explored the blockchain technology as a prospective solution [15,17–19]. In addition, researchers have effectively deployed consortium blockchain to enhance security and privacy in vehicular network information dissemination. For instance, Kang et al. [19] introduced a secure sharing framework based on the consortium blockchain and smart contract technologies. This approach employs a tri-weight subjective logic model to elect data sources, enhancing the credibility of the disseminated data. Chen et al. [20] introduced a trust computation methodology based on collaborative filtering. Using this technique, they considered inter-vehicular interactions and historical data to accurately infer and evaluate trust values, thus enhancing the credibility of the information dissemination. Javaid et al. [21] have employed physical unclonable functions (PUF) and certificates in blockchain protocols, facilitating faster trust management among vehicles. Zhang et al. [22] then established a blockchain-based trust management mechanism to preserve vehicular reputation values. This research enhanced inter-vehicular trust management, bolstering secure information dissemination in vehicular networks. Furthermore, provided by these studies is a solid theoretical foundation and practical guidance for security and trust management in Telematics, which not only demonstrate the potential of blockchain technology in addressing the problems traditional to the field but also experimentally prove its effectiveness in enhancing the security and trustworthiness of data. Yang et al. [23] suggested a decentralized trust management system using Bayesian inference models to validate information, further supporting secure communication. El-Sayed et al. [24] estimated trustworthiness among vehicular entities using a hybrid trust model, ensuring secure and reliable information exchange. Collectively, these studies underscore the potential of blockchain technology in fortifying the security and trustworthiness of vehicular networks.

C. Challenges to Blockchain Technology

Due to its enhanced efficiency and reduced costs, consortium blockchain emerges as a superior choice for scenarios requiring secure vehicular data sharing. Cui et al. [18] introduced a consortium blockchain-based scheme for secure vehicular data sharing, which facilitates efficient traceability and reliable information sharing. They also conceptualized a consensus algorithm based on a trust score model, aimed at improving the quality and reliability of the shared data. Despite the blockchain's significant advantages, such as secure vehicular data sharing and effective mitigation of information tampering [18–20,22], its immutable mechanism poses challenges. Once data has been appended to a block, altering or removing it becomes difficult. This issue could potentially arise in situations such as erroneous data entry or unintended disclosure of confidential information. Furthermore, blockchain nodes are required to store substantial data, leading to high storage demands and latency issues [25]. This is particularly essential in vehicular networks, where vehicles constantly generate and exchange data. Utilizing blockchain technology for data processing could potentially affect the network's stability and latency [26]. Qiao et al. [27] emphasized the importance of evaluating both direct and indirect trust, proposing a combined trust evaluation model involving edge servers and data users to improve trust assessment accuracy. Therefore, while blockchain offers promising solutions, its implementation in vehicular networks requires careful consideration of these challenges.

To address the issues outlined above, this paper proposes a trust aggregation authentication scheme, through a trust scoring system model for vehicles. The scheme tackles malicious vehicles dissemination, the single point of failure in centralized trust models, and the high communication and storage overhead in blockchain-based vehicular networks.

3 System Model and Attack Model

3.1 System Model

The proposed model for preserving vehicular privacy through a trust-score system is depicted in Fig. 1. It is composed of CSSs arrays, TAI, RSUs, and Vehicles (V). The roles of each component are explained below:

CSSs:

They are primarily responsible for monitoring traffic conditions in real-time and collating relevant information. In practice, this responsibility is delegated to the camera-based traffic monitoring system. Leveraging advanced sensors and communication devices, CSSs can capture complex vehicular motion parameters such as location, velocity, and acceleration. CSSs are fully trusted entities capable of withstanding various attacks. They possess ample storage resources and computational power, enabling them to monitor vehicular information promptly and accurately. This capability ensures that traffic conditions are continuously monitored, and relevant data is reliably collected and processed in real-time.

TAI:

This component consists of official entities such as government transportation departments. Its duties include generating public-private key pairs, issuing public keys, and registering vehicular identity information. When a vehicle detects malicious behavior from another, it can report this to the TAI. Subsequently, TAI conducts an initial screening and verification of the reported information before collaborating with CSS to verify the accuracy and timeliness of the information.

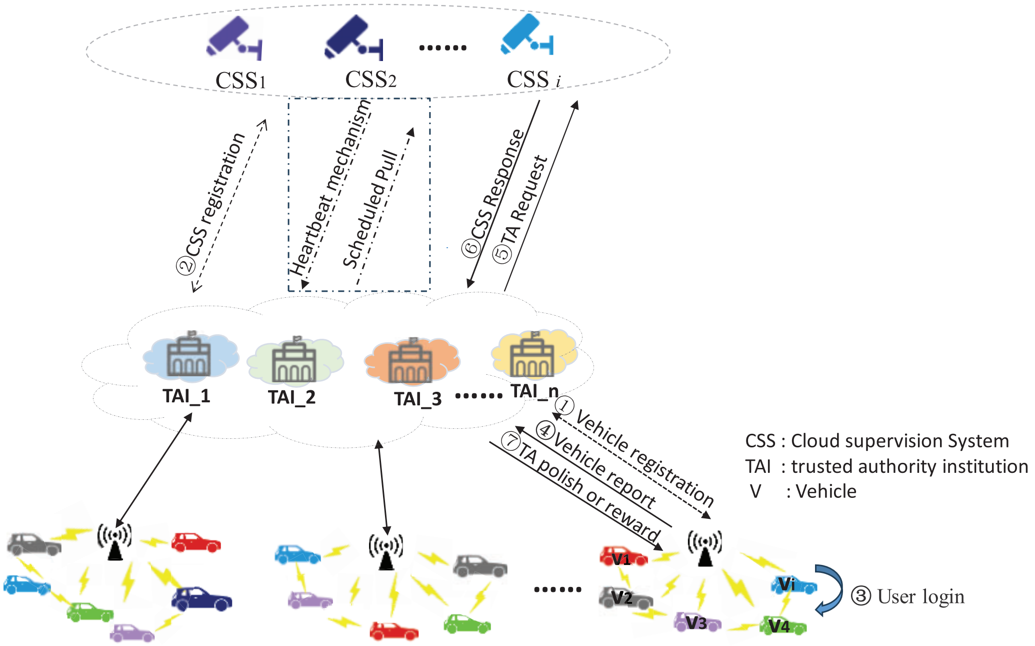


Figure 1: System model

V:

They are equipped with communication devices, including Dedicated Short Range Communication (DSRC) and Wireless Access for Vehicular Environment (WAVE), which facilitate communication with their surroundings [28]. These devices can receive and transmit data such as the vehicle's location, velocity, direction, road conditions, and traffic signals. These data provide essential information about the vehicle's status and the environmental conditions. Simultaneously, each vehicle incorporates an On-Board Unit (OBU) that serves as the central controller, interfacing with the Electronic Control Unit (ECU) to manage and control the vehicle's operations.

RSUs:

They are wireless communication devices engineered to enhance intelligent transportation systems. They enable bidirectional data exchange with Cluster Head Vehicles (CHVs), solely relaying CHVs' information to the TAI. In our scheme, different regions are divided by various RSUs, with interactions between regions conducted through CHVs. The formation of these regions generally utilizes scalable clustering algorithms [29]. It is noteworthy that the selection of clustering algorithms is not the focus of this paper. If you are interested in this topic, you can refer to References [30–33]. Vehicles entering a new region cannot access information from their previous region or the historical data of the new region, ensuring data security. This facilitates the identity verification of vehicles and data encryption without storing any personal information of the users or participate in the authentication operation. Moreover, it functions solely as a relay node. This fundamentally eliminates the risk of crucial data breaches, thus safeguarding the privacy and security of vehicle users.

3.2 Attack Model

The wireless interaction between V and CSS poses the risk of interception by malicious entities. These attackers may exploit vulnerabilities to steal, manipulate, or engage in impersonation and replay attacks. To address this risk, we employ the established Canetti and Krawczyk's adversary model (CK-adversary model) [34] as our threat framework. Specifically, in this study, RSU serve solely as intermediaries for message forwarding, without engaging in message storage; the TAI acts as the authenticating body for messages between V and CSS. We make the assumption that both RSUs and TAI are reliable.

3.3 Security Assumption

The distributed vehicle system model based on trust scoring proposed in this paper must be resistant to various malicious attacks while maintaining reliable performance. The specific requirements are as follows:

(1) *Anonymity*: It is required to prevent other entities from recognizing the true identity of the vehicle user. Thus, it is necessary to effectively encrypt the identity of the vehicle user during the interaction process to ensure that his or her identity is not compromised.

(2) *Traceability*: In vehicular communication systems, vehicles exchange information to enable functions such as real-time traffic information sharing and real-time road condition monitoring. To ensure the source and authenticity of the messages, it is crucial that messages sent by vehicles are traceable. Failure to trace messages can pose a potential threat to traffic safety.

(3) *Mutual Authentication*: It refers to the bidirectional identity verification between vehicles, which aims to confirm the identity and legitimacy of the communicating vehicles. This helps prevent the intrusion and attacks from malicious entities, and ensures communication only with authorized vehicles.

(4) *Resistance to Data Replay Attack*: It occurs when an attacker intercepts and replays legitimate vehicle control commands, leading to misleading and potentially dangerous vehicle operations. Furthermore, the attacker can manipulate location information and road condition data, disrupting the cooperative work and decision-making processes of the vehicle network, and potentially resulting in a chaotic vehicle network system. To mitigate this attack, the inter-entity interaction verification is employed in this paper.

(5) *Timed Heartbeat Mechanism and Pull Services*: To maximize utility, CSS entities are scheduled to send availability messages to TAI before a vehicle sends a request. Additionally, TAI is scheduled to pull the list of CSS services to verify their availability.

4 The Proposed Scheme

4.1 System Overview

In this section, we will introduce the proposed scheme from four phases: 1) System initialization phase; 2) Registration phase; 3) User login phase; and 4) Trust score update phase. Table 1 shows the notations of the proposed scheme.

4.2 System Initialization Phase

In this phase, the TAI_i selects a prime additive cyclic group G, where the operation of $G \times G = G^v$, and the order is denoted as P. TAI_i generates a public-private key pair (S_i, P_i) where $P_i = S_i P$ and S_i is a randomly generated private key. Then, it generates two secure hash functions, $H_1 : \{0, 1\}^* \rightarrow \{0, 1\}^v$, and $H : \{0, 1\}^* \rightarrow G^v$. TAI_i exposes system parameters $\{H_1, H, P_i, P\}$.

Table 1: Notations

Notation	Description
$\boldsymbol{CSS_i}$	Node of the $i-th$ Cloud supervision system
$\boldsymbol{TAI_i}$	The $i-th$ trusted institution node
$\boldsymbol{P_i/S_i}$	Public/Private Key of TAI_i
$\boldsymbol{C_{pub_i}}$	Public Key of the $i-th$ vehicle
$\boldsymbol{C_{pri_i}}$	Private Key of the $i-th$ vehicle
$\boldsymbol{UID_i}$	Real identity of $User_i$
$\boldsymbol{AID_i}$	Anonymous identity of $User_i$
$\boldsymbol{VID_i}$	Real identity of V_i
$\boldsymbol{CID}$	Real identity of CSS_i
$\boldsymbol{UPW_i}$	The password of $User_i$
$\boldsymbol{V_i}$	The $i-th$ vehicle
$\boldsymbol{CT_i}$	The current timestamp
$\Delta\boldsymbol{CT}$	Maximum threshold for timestamp validity
$\boldsymbol{H, H_1}$	Two one-way hash functions
$\boldsymbol{r, s, t, u, v}$	Random numbers
$\oplus, \vert\vert$	Exclusive OR and concatenation operation
$\boldsymbol{TS_i}$	Trust scores of V_i
$\boldsymbol{r_i}$	The $i-th$ cluster head vehicle request message
$\boldsymbol{TTS}$	The threshold for the trust score is equal to 69
$\boldsymbol{VNUM}$	Number of vehicles reported
$\boldsymbol{CHV_i}$	The $i-th$ cluster head vehicle
$\boldsymbol{LTV/HTV}$	Lower/Upper threshold value for the number of vehicles reported
$\boldsymbol{RTE/PTE}$	Reward/Penalty trust score threshold

4.3 Registration Phase

To participate in the trusted traffic system, V_i and CSS_i must register and establish secure communication with TAI_i to obtain authorizations and certificates. Upon successful registration, the TAI_i verifies the credentials and issues digital certificates to ensure secure and authenticated interactions within the vehicular network.

4.3.1 Vehicles Registration

Registration of the vehicle is essential for accessing various services and ensuring personal safety. Fig. 2 schematically represents the vehicle registration process.

(1) Initially, the user $User_i$ is required to provide a unique identity identifier UID_i to TAI_i. Concurrently, $User_i$ randomly selects a password UPW_i, and computes $RIW_i = H_1(UID_i \oplus UPW_i)$. Afterward, Subsequently, $User_i$ sends TAI_i both UID_i and RIW_i through the vehicle for identity verification.

(2) Upon receiving the parameters, TAI_i first queries the backend data via UID_i to determine whether the current user's vehicle is registered. For a new user, TAI_i generates a random number s_i, and calculates $A_i = RIW_i \oplus H(VID_i||s_i)$. Then, it sends $\{A_i, H_1, H\}$ to the vehicle user, stores them in the vehicle system, and generates a public-private key pair. The user $User_i$ then generates a unique random number r, computes $K_i = P_i * (AID_i \oplus RIW_i)||r$, $Q_i = K_i \oplus RIW_i$, $B_i = P_i * H_1(RIW_i||VID_i||K_i)$, and stores $\{B_i, Q_i\}$ in the vehicle storage's unit.

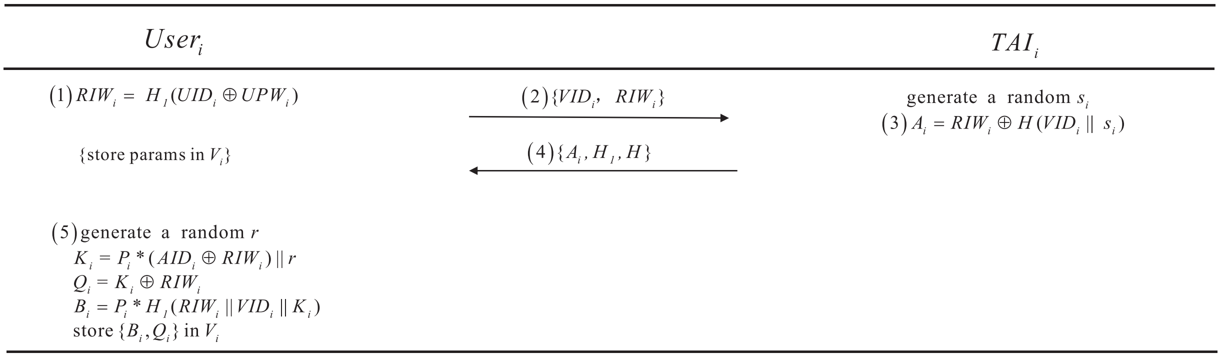

$User_i$
TAI_i
(1) $RIW_i = H_1(UID_i \oplus UPW_i)$
(2) $\{VID_i, RIW_i\}$
generate a random s_i
(3) $A_i = RIW_i \oplus H(VID_i \| s_i)$
{store params in V_i}
(4) $\{A_i, H_1, H\}$
(5) generate a random r
$K_i = P_i * (AID_i \oplus RIW_i) \| r$
$Q_i = K_i \oplus RIW_i$
$B_i = P_i * H_1(RIW_i \| VID_i \| K_i)$
store $\{B_i, Q_i\}$ in V_i

Figure 2: Vehicle registration phase

4.3.2 Cloud Supervision System Registration

Fig. 3 depicts the interaction process between the CSS and TAI. Each service provider in CSS, identified as CSS_i, is required to register with TAI_i by submitting its identity information and relevant credentials for verification and storage.

(1) CSS_i initially employs the pseudo random number generator (PRNG) algorithm [35] to generate random numbers u, CID_i, and x_i . Subsequently, it computes $D_i = H_1(CID_i \oplus u)$, and transmits the parameter D_i to TAI_i.

(2) After receiving the parameters from CSS_i, TAI_i generates a random number v that is independent and unpredictable. Then, it computes $E_i = D_i \oplus v$, $E_i = \alpha_i P$, where α_i is a random number generated by TAI_i. Afterwards it sends E_i and U_i to CSS_i, and stores $\{E_i, U_i\}$ in a secure database.

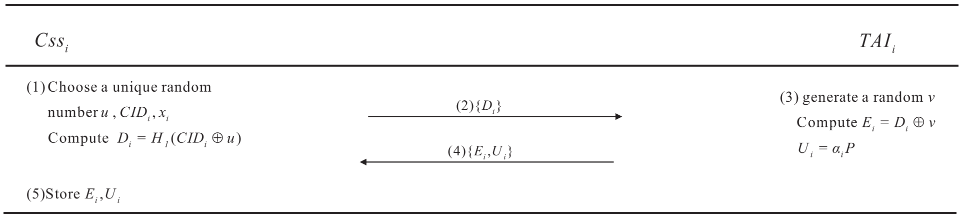

Css_i
TAI_i
(1) Choose a unique random
number u, CID_i, x_i
Compute $D_i = H_1(CID_i \oplus u)$
(2) $\{D_i\}$
(3) generate a random v
Compute $E_i = D_i \oplus v$
$U_i = \alpha_i P$
(4) $\{E_i, U_i\}$
(5) Store E_i, U_i

Figure 3: Cloud supervision system registration phase

4.4 User Login Phase

Fig. 4 illustrates the steps for a user to log into the vehicle system. This process ensures the legitimacy of the user's identity and the system's security.

(1) $User_i$ inputs his or her name and password. Next, he sends $\{UID_i, UPW_i\}$ to the V_i.

(2) After receiving the message from $User_i$, the vehicle verifies whether the entered password is correct. V_i computes $C_i = H_1\,(UID_i \oplus UPW_i)$, $K_i = C_i \oplus Q_i$,$B_i^{'} = P_i{*}H_1\,(C_i||UPW_i)$, and compares it with B_i. If $B_i^{'} = B_i$, then the user's login is successful and a verification success message is returned. Otherwise, the user's login fails, and a verification failure message is returned.

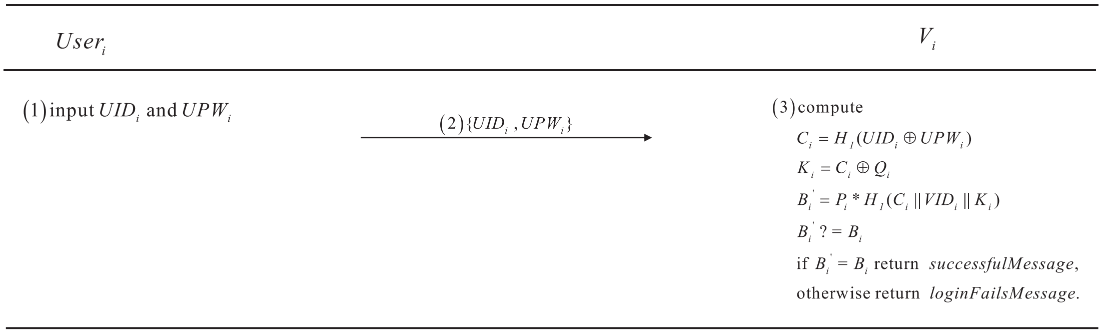

$User_i$
V_i
(1)input UID_i and UPW_i
(2)$\{UID_i, UPW_i\}$
(3)compute
$C_i = H_1(UID_i \oplus UPW_i)$
$K_i = C_i \oplus Q_i$
$B_i^{'} = P_i * H_1(C_i \| VID_i \| K_i)$
$B_i^{'} ?= B_i$
if $B_i^{'} = B_i$ return $successfulMessage$,
otherwise return $loginFailsMessage$.

Figure 4: User login phase

4.5 Trust Score Update Phase

Legitimate vehicles send report information to TAI_i, initially verifying the vehicle user's identity and then querying a specific CSS to validate the integrity of the information. A CSS node facilitates TAI_i in adjusting its decision-making and response in real-time. Specifically, the CSS node processes and aggregates the received data, performing an initial integrity check before forwarding it to TAI_i for a more comprehensive analysis. Users' information is encrypted for security. Upon decryption, TAI_i audits each CSS node to ensure the provision of service, analyzing node performance, availability, and load conditions.

4.5.1 Inter Vehicle Authentication and Key Agreement

(1) First, the vehicles are required to establish initial trust by exchanging messages (m_i) and establishing a session key [26]. The vehicles, v_i and v_j, assess and score each other's messages before forwarding them to CHV_i. Here, the CHV_i is elected through the evaluation of trust scores from vehicles in the domain at predetermined intervals. Any vehicle within a certain area of the RSU may be elected as a cluster head vehicle. This process is shown in Fig. 5.

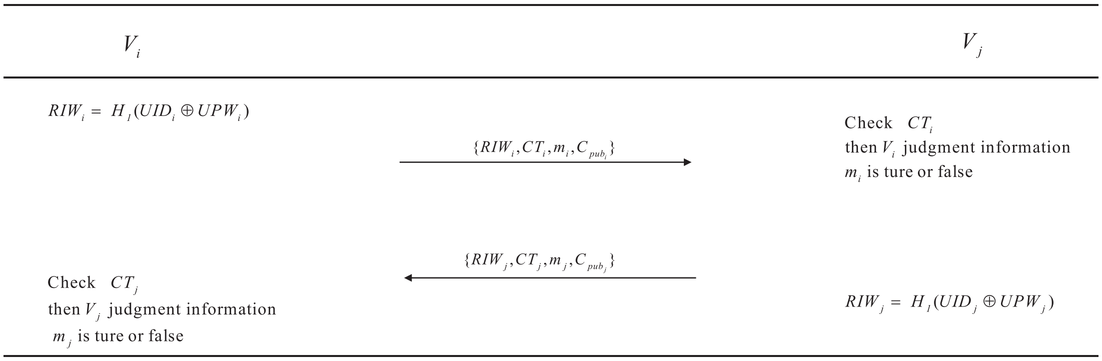

V_i
V_j
$RIW_i = H_1(UID_i \oplus UPW_i)$
$\{RIW_i, CT_i, m_i, C_{pub_i}\}$
Check CT_i
then V_i judgment information
m_i is ture or false
$\{RIW_j, CT_j, m_j, C_{pub_j}\}$
Check CT_j
then V_j judgment information
m_j is ture or false
$RIW_j = H_1(UID_j \oplus UPW_j)$

Figure 5: Inter vehicle authentication and key agreement

(2) To be specific, v_i calculates $RIW_i = H_1\,(UID_i \oplus UPW_i)$ and encrypts it using v_j's public key C_{pub_j}. Then, v_i sends the parameter $\{RIW_i, M_i, CT_i, C_{pub_i}\}$ to v_j. After receiving the message from v_i, v_j decrypts it with his private key to assess its correctness, score v_i, and rank trust scores of vehicles in the domain. The scoring information are then sent to CHV_i, the vehicle with the highest trust score in the domain. CHV_i forwards the scored distribution to v_i and stores it. Similarly, v_j establishes a session key with v_i in the same manner.

(3) The delivery rate (DR) [36], effective propagation rate (EPR), and safety-distance information rate ($SDIR$) [37] are the three key metrics used to calculate a vehicle's trust score. The higher these three metrics are, the more reliable the vehicle is and the higher the score. Conversely, vehicles with low metrics are indicative of serious transmission failures during data transmission. Therefore, the trust level of the affected vehicle should be adjusted downwards to reflect its decreased reliability.

4.5.2 Trust Score Update

In the initialization phase, vehicles obtain initial trust through mutual interactions and historical behavior information. To accurately evaluate the trustworthiness of vehicles, our model employs three methods: DR, EPR, and $SDIR$. When a vehicle's malicious behavior is reported by another vehicle, the report is first preliminarily screened by the TAI and subsequently verified by the CSS. In this interaction, the CSS is responsible for validating the integrity of the report by comparing it with locally stored data and logs, and then sending its verification results back to TAI for final judgment. If the report is found to be accurate, the TAI rewards the reporting vehicle and penalizes the malicious vehicle by deducting trust points. Conversely, if the report is false and the reported vehicle was wrongly accused, the CSS verifies this, and the vehicle that was falsely accused is rewarded with trust points, while the malicious reporting vehicle is penalized.

(1) The vehicle trust may increase or decrease, depending on the vehicle's behavior of the vehicle. As illustrated in Fig. 6, vehicle V_i first computes $RIW_i = H_1\,(UID_i \oplus UPW_i)$ and selects a random number f_i. The $User_i$ verifies his identity by logging into the system using the pre-stored static parameters UID_i and UPW_i. Subsequently, the system calculates $G_i = f_i P$, $G_i^* = f_i P_i$, $F_i = RIW_i \oplus G_i^* \oplus A_i$, and $M_i = H_1\,(VID_i || CT_i) \oplus r_i$, respectively. To enhance the efficiency of transmitting vehicle report information and reduce communication overhead, the system utilizes CHV_i as an intermediary node for transmitting the vehicle report information r_i. Finally, the parameters $D1$ are sent to TAI via the network.

(2) Upon receiving the requested data from the vehicle, TAI_i calculates whether $\Delta CT < |CT_r - CT_i|$ to ensure the data's timeliness in generation or transmission. If CT_i is valid, TAI_i decrypts the data from V_i and computes $G_i^* = S_i G_i$. Then, it checks the validity of $RIW_i = F_i \oplus G_i^* \oplus A_i$. Since RIW_i is authenticated at registration, matching the received RIW_i demonstrates the validity of the equation, allowing TAI_i to proceed with subsequent operations. After verification, TAI_i encapsulates the validated data and associated parameters before securely transmitting parameters $D2$ to CSS_i for further processing, ensuring that CSS_i can accurately assess and aggregate the data for system-wide decisions.

(3) In Spring Cloud Netflix [38], the registry Eureka [39] is designed to ensure high service availability. In this paper, we adopt a similar approach, using CSS_i as a registry to guarantee effective service provision. Through mutual registration, each CSS_i maintains a registry list to track the availability of other CSS_i. Periodically, TAI_i queries the CSS_i registry to check service availability. Unavailable services are removed from the CSS_i registry. Furthermore, CSS_i periodically sends heartbeats to TAI_i. If the heartbeat intervals exceed the specified threshold, CSS_i is marked as unavailable. To prevent premature removal, TAI_i automatically activates the protection mechanism

based on the CSS_i heartbeat ratio. Specifically, TAI_i checks whether the heartbeat ratio of CSS_i falls below the predefined threshold at regular interval. If the ratio meets the threshold, CSS_i remains in service.

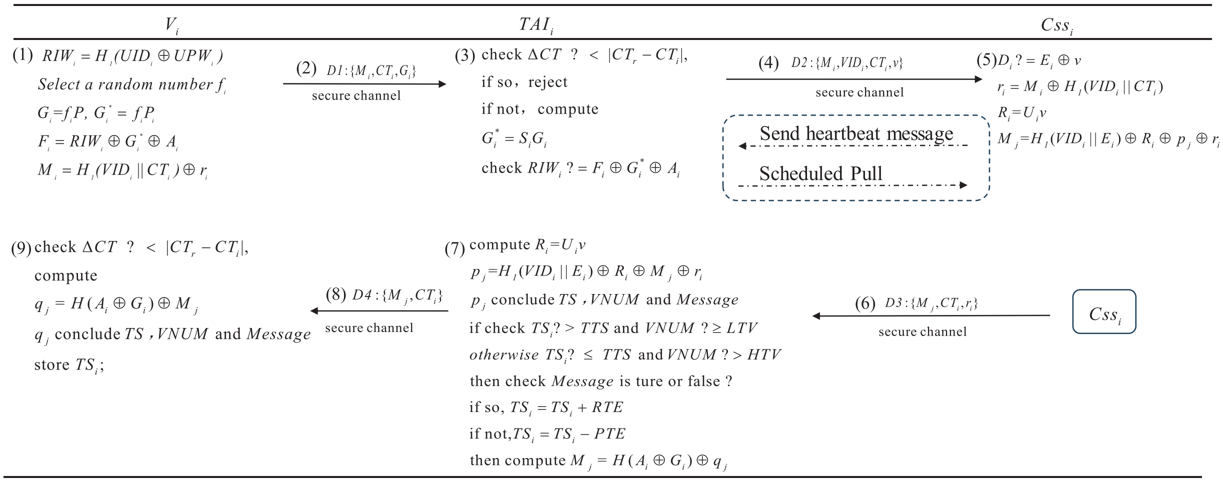

V_i
TAI_i
Css_i
(1) $RIW_i = H_1(UID_i \oplus UPW_i)$
$Select\ a\ random\ number\ f_i$
$G_i = f_iP,\ G_i^* = f_iP_i$
$F_i = RIW_i \oplus G_i^* \oplus A_i$
$M_i = H_1(VID_i \| CT_i) \oplus r_i$
(2) $D1:\{M_i, CT_i, G_i\}$
secure channel
(3) check $\Delta CT\ ?<|CT_r - CT_i|$,
if so，reject
if not，compute
$G_i^* = S_iG_i$
check $RIW_i\ ?= F_i \oplus G_i^* \oplus A_i$
(4) $D2:\{M_i, VID_i, CT_i, v\}$
secure channel
Send heartbeat message
Scheduled Pull
(5) $D_i\ ?= E_i \oplus v$
$r_i = M_i \oplus H_1(VID_i \| CT_i)$
$R_i = U_iv$
$M_j = H_1(VID_i \| E_i) \oplus R_i \oplus p_j \oplus r_i$
(6) $D3:\{M_j, CT_i, r_i\}$
secure channel
Css_i
(7) compute $R_i = U_iv$
$p_j = H_1(VID_i \| E_i) \oplus R_i \oplus M_j \oplus r_i$
p_j conclude TS，$VNUM$ and $Message$
if check $TS_i? > TTS$ and $VNUM\ ? \geq LTV$
$otherwise$ $TS_i? \leq TTS$ and $VNUM\ ? > HTV$
then check $Message$ is ture or false ?
if so, $TS_i = TS_i + RTE$
if not, $TS_i = TS_i - PTE$
then compute $M_j = H(A_i \oplus G_i) \oplus q_j$
(8) $D4:\{M_j, CT_i\}$
secure channel
(9) check $\Delta CT\ ?<|CT_r - CT_i|$,
compute
$q_j = H(A_i \oplus G_i) \oplus M_j$
q_j conclude TS，$VNUM$ and $Message$
store TS_i;

Figure 6: Phase of updating trust scores

Algorithm 1: Heartbeat mechanism

```
Input: $CSS_{rList}$: $\{CSS_1, CSS_2, \cdots, CSS_i, \cdots, CSS_k\}$
Output: $CSS_{nRList}$: $\{CSS_1, CSS_2, \cdots, CSS_i, \cdots, CSS_m\}$
$TAI_i$: for ($i = 0,, i < CSS_{rList}$.length, $i++$)
        check $CSS_i$ is $normal$
        then renew ($TCSS_i$) $CSS_{i_{hc}}++$
         $CSS_i$ is $abnormal$ then $CSS_{i_{Count}}++$
            If($CSS_{i_{hc}}$ && $CSS_{i_{Count}} > HVALUE$)
            else logger.info("$please\ register$")
            end if
      end for
      $CSS_{nRList} = LB$(sort ($CSS_{rRList}$))
```

To ensure the high availability of CSS_i, we utilize a round-robin load balancing algorithm (see Algorithm 1) to select the CSS_i service. Since the choice of load balancing algorithm is beyond the scope of this paper. To gain a deeper understanding, reference to the related literature [40–43] is recommended.

TAI determines the available services list (CSS_{rList}) by analyzing the service registry and generates a new list (CSS_{nRList}) of available services after applying using a load balancing algorithm **(**LB**)** [41]. After evaluating the availability of each service, TAI updates the service renewal time ($TCSS_i$) and increases the heartbeat count ($CSS_{i_{hc}}$) if the service is available. If the service's heartbeat count and the number of TAI requests for CSS service availability ($CSS_{i_{Count}}$) both exceed the threshold ($HVALUE$), the service is considered available. Otherwise, the service needs to be re-registered. Finally, the service is reordered and made available based on the weights and heartbeat times.

(4) CSS_i checks whether D_i is equal to $E_i \oplus v$. If it holds, then CSS_i is authenticated as a legally registered entity in TAI_i. It proceeds to calculate the message $r_i = M_i \oplus H_1(VID_i||CT_i)$, $R_i = U_iv$,

$M_j = H_1(VID_i||E_i) \oplus R_i \oplus p_j \oplus r_i$, where p_j is a judgment message used by the CSS to verify the accuracy of the reported information from the vehicle, and the number of trust scores and reported vehicles that each vehicle needs to update. In this process, CSS_i also evaluates the trustworthiness of the report based on data received from TAI_i, ensuring that the information aligns with previously registered vehicle behavior. This evaluation is performed before generating and sending the final message, incorporating the trust analysis as part of its authentication workflow. Finally, CSS_j transmits the computed message and parameters $D3$ back to TAI_i.

(5) CSS_i sends parameters CT_i to TAI_i, who then checks whether the timestamp of the sent information is within the specified delay time. If it is valid, TAI_i compute $R_i = U_i v$, and $p_j = H_1(VID_i||E_i) \oplus R_i \oplus M_j \oplus r_i$. Subsequently, it judges whether the inequality $TS_i > TTS$ holds, where TS_i represents the trust scores of all reporting vehicles. If this condition holds, it indicates that the trust score of the reporting vehicle exceeds the specified threshold. It also checks whether $VNUM \geq LTV$ holds. If true, this indicates that $VNUM$, the number of reporting vehicles exceeding the trust score threshold, meets or exceeds the minimum threshold. If any condition is not met, it judges whether $TS_i \leq TTS$ and $VNUM \geq HTV$ hold. These conditions indicate that the trust score of the reporting vehicle is at most equal to the specified threshold and the number of reporting vehicles exceed the maximum threshold. If the number of reporting vehicles and the trust score meet the above judgments, reporting is allowed; otherwise, it is not. Subsequently, TAI_i rewards or penalizes the reporting and reported vehicles based on the accuracy of the message from CSS_i. If the report information is true, TAI_i applies the equation $TS_i = TS_i + RTE$ to reward the reporting vehicle, and calculates $TS_i = TS_i - PTE$ to penalize the reported vehicle that spreads malicious information. Otherwise, it penalizes the reporting vehicle and rewards the reported one for spreading correct information, using trust score measures. Finally, TAI_i computes $M_j = H(A_i \oplus G_i) \oplus q_i$, where q_i includes scores of reporting and reported vehicle entities, as well as the relevant reporting data of vehicles.

Algorithm 2: Vehicle rewards and punishments

```
**Input:** Calculated the initial trust scores: $\{TS_{Initial_1}, TS_{Initial_2}, \cdots, TS_{Initial_i}, \cdots, TS_{Initial_k}\}$
**Output:** $TS = \{TS_1, TS_2, \cdots, TS_i, \cdots, TS_k\}$
$CHV_i = scheduledSort(TS_1, TS_2, \cdots, TS_i, \cdots, TS_n)$
$V_i \rightarrow TS_{V_j}$ to $CHV_i$,  $CHV_i$ compute $TS_{Initial_j} += RTE$ or $TS_{Initial_j} -= PTE$;
$V_j \rightarrow TS_{V_i}$ *to* $CHV_i, CHV_j$ compute $TS_{Initial_i} += RTE$ or $TS_{Initial_i} -= PTE$
  $TS_i = TS_{Initial_i}$,  $TS_j = TS_{Initial_j}$
  $m_i = reportV(V_1, V_2, \cdots, V_i, V_j) \rightarrow CHV_i$
  $CHV_i \rightarrow TAI_i, TAI_i \rightarrow CSS_i$
  $CSS_i$: **Check** $m_i$ is True or False **then** $\rightarrow TAI_i$
  $TAI_i$: compute
  **for** $(i = 0, , i < rtV_{v_i}$.**length**, $i++)$
    **if** $(rtV_{v_i}.TS_i > TTS \&\& VNUM \geq LTV)$
        **if**($rtV_{v_i}$ is True and $rtedV_{v_i}$ is False)  $TS_{rtV_{v_i}} += RTE, TS_{rtedV_{v_i}} -= PTE$
        **else if**($rtV_{v_i}$ is False and $rtedV_{v_i}$ is True) $TS_{rtV_{v_i}} -= PTE, TS_{rtedV_{v_i}} += RTE$
        **end if**
    **else if** $(rtV_{v_i}.TS_i \leq TTS \&\& VNUM > HTV)$
        **if** ($rtV_{v_i}$ is True and $rtedV_{v_i}$ is False)  $TS_{rtV_{v_i}} += RTE$, $TS_{rtedV_{v_i}} -= PTE$
        **else if**($rtV_{v_i}$ *is* False and $rtedV_{v_i}$ is True)  $TS_{rtV_{v_i}} -= PTE, TS_{rtedV_{v_i}} += RTE$
        **else logger.info** ("Report Vehicle Lack of Confidence")
```

(Continued)

Algorithm 2 (continued)

```
            end if
        end if
end for
```

Primarily, the initial trust scores ($TS_{Initial_i}$) of vehicles are computed using three indicators: DR, EPR, and $SDIR$. Vehicles V_i and V_j mutually assign scores, represented as TS_{V_j} and TS_{V_i}, respectively. Subsequently, CHV_i periodically uses the *scheduledSort* function to select the RSU regional vehicle trust score (see Algorithm 2). Then, V_i and V_j each transmit their scores to CHV_i individually. Based on real road conditions, CHV_i uses $TS_{Inital_i} + = RTE$ to increase the trust score for the vehicle transmitting information, and $TS_{Inital_i} - = PTE$ to decrease it.

Ultimately, CHV_i calculates $TS_i = TS_{Initial_i}$. Upon detecting malicious vehicles disseminating illegal information, multiple vehicles are required to report to CHV_i ($reportV$). Subsequently, CHV_i relays the information m_i to TAI_i. In each region divided by RSUs, a CHV_i can be selected to ensure security. When a vehicle enters a new region, it cannot access information from the previous region nor the historical data of the new region.

In Algorithm 2, the TAI_i transmits m_i to CSS_i. CSS_i assesses the accuracy of m_i, including the data from the reporting vehicle (rtV_{v_i}) and the reported vehicle ($rtedV_{v_i}$), and then reports the results to TAI_i. If the condition $rtV_{v_i}.TS_i > TTS\&\&VNUM \geq LTV$ or $rtV_{v_i}.TS_i \leq TTS\&\&VNUM > HTV$ is satisfied, TAI_i adjusts the trust score based on the accuracy of the information from both vehicles, either as a reward or a penalty. If neither condition is met, indicating distrust, a “Report Vehicles Lack of Confidence” log is generated. In this case, the rtV_{v_i} needs to successfully report more instances, thus increasing its trust score.

(6) TAI_i dispatches the calculated trust scores for the reporting and reported vehicles, along with the current timestamp, to the CHV_i. CHV_i checks whether $\Delta CT < |CT_r - CT_i|$ holds, and then computes $q_i = H(A_i \oplus G_i) \oplus M_j$. Before this, TAI_i retrieves the necessary context from the CSS, including any relevant cluster information and authentication parameters, ensuring the calculation of accurate trust scores. Finally, CHV_i transmits the trust score to the corresponding V_i, for storage in the vehicle system. It should be noted that, upon reelection of CHV_i, the trust score of the associated vehicle is synchronized with the latest cluster head vehicle, and locally stored trust scores of other vehicles are deleted. This process safeguards the vehicle’s security.

5 Security Proof and Analysis

This section mainly analyzes and proves the security of the proposed authentication scheme.

5.1 Security Proof

5.1.1 Security Model

To ensure secure interactions between V and CSS, we have developed a robust security model. Our goal is to ensure the system’s reliability and confidentiality. The model involves three entities: V, CSS, and TAI. We denote their arbitrary instances as V_a, CSS_b, and TAI_c. Given that Section 3 explicitly establishes TAI’s absolute security and RSU’s role as a mere forwarder (without information storage), we exclude them as potential threats in our analysis.

(1) *AKA Request*: In this scenario, adversary (A) intercepts the information exchanged during the authentication and key negotiation phase among entities V_a, CSS_b, and TAI_c. This process is vulnerable

to a potential security threat of eavesdropping, as A attempts to listen to on the communication between V_a, CSS_b, and TAI_c. At this juncture, challenger (C) responds by generating pseudo-data of equivalent size.

(2) *Indirect Request*: During this communication, A attempts to intercept and manipulate the message between the parties. In response to this tampering threat, C promptly addresses the situation by replying within the same context.

(3) *Private Key Request*: In the context of a communication session, A initiates a request to query the session key shared among participating entities. Upon the session's completion, C securely transmits the session key to A by encrypting the message, and then intercepting a portion of the encrypted transmission.

(4) *Test Request*: A initiates a query to retrieve interaction information. In response, C assesses the information's freshness based on the timestamp and validates it through entity encryption. Finally, C returns the appropriately formatted information.

Definition1: Considering the assumption that the successful outcome of A in the game is a polynomial-time problem (PPT), we can conclude that our proposed scheme is deemed to possess absolute security.

Definition2: If the interaction among V_a, CSS_b, and TAI_c meets the requirement for the initial session visit within the specified short timestamp, and if it is computationally challenging to derive the interaction information using DHP, we additionally employ encryption or pseudorandom data generation to ensure the security of our proposed scheme.

5.1.2 Security Proof

The scheme proposed in this paper relies on Diffie-Hellman Challenge (DCH) and Diffie-Hellman Problem (DHP), as below:

Theorem: In response to A's parameterized request attack, our scheme utilizes the approach of creating pseudo-data resembling the original and validating freshness, thereby guaranteeing the scheme's efficacy.

Proof: To assess the security of our proposed scheme, we introduce a game between challengers C and opponents A. It is crucial to evaluate any potential vulnerabilities If A can compromise our scheme. During the game, A may initiate the following requests:

Initialization: C Setup List: $List_1 = \left\{RIW_i, m_i, C_{pub_i}, S_{pri_i}\right\}, List_2 = \{A_i, CIW_i, RIW_i\}, List_3 = \{VID_i, M_i, v, e, CT_i\}, List_4 = \{CT_i, RIW_i\}, andList_5 = \{P_i, G_i, r\}$. In this phase, C initializes all of the aforementioned lists with empty values.

(1) *AKA Request*: When A seeks to inquire about the messages (M_1, M_2, M_3, M_i) exchanged between V_a, CSS_b, and TAI_c, C then generates random data of equivalent size to the message. Subsequently, C encrypts this random data by computing G_i and by P_i, storing it in $List_5$ before transmitting the message to A.

(2) *Indirect Request*: In the given context, A sends a request for M_i to TAI_c. Subsequently, C initially sends M_i to TAI_c by computing $M_i = H_1\left(VID_i || CT_i\right) \oplus r_i$ and establishing the current timestamp. Following this, C sends M_i to TAI_c.

(3) *Private Key Request*: When A makes a query with RIW_i, C verifies the presence of $(RIW_i, m_i, C_{pub_i}, S_{pri_i})$ in $List_4$. If there exists a session key S_{pri_i} in $List_1$, C computes $H(S_{pri_i})$, extracts

data with a byte size equivalent to S_{pri_i}, stores it in $List_1$, and subsequently sends the information back to A.

(4) *Test Request*: Upon receiving a request with CT_i from A, C evaluates the freshness of the timestamp using formula $\Delta CT? < |CT_r - CT_i|$. If the data is fresh, C then records the interaction information (M_1, M_2, M_3, M_i) between them. Subsequently, the equation $G_i = f_i P$ is utilized for computation, and the resulting data is stored in $List_5$. Finally, the data from $List_5$ (M_i, CT_i, G_i) is transmitted to A.

5.2 Security Analysis

(1) *Anonymity*: In the vehicular network discussed in this paper, protecting the identity and password of vehicle users from potential theft is considered crucial. To achieve this objective, a hash function derived as $RIW_i = H_i (UID_i \oplus UPW_i)$ is used to encrypt the user's identity and password during interactions with other entities.

(2) *Traceability*: The report message sent by each vehicle is attached with its unique identification V_i. When vehicles interact, they calculate unique encrypted information based on each vehicle's unique identity, ensuring the sender's identity and the message's traceability to TAI_i. For each message, CHV_i calculates $M_i = H_1 (VID_i || CT_i) \oplus r_i$, records the sender, and transmits the message to TAI_i. These steps accurately identify vehicles during transmission, preventing tampering or impersonation.

(3) *Mutual authentication*: Messages from vehicle V_i to vehicle V_j are encrypted with V_j's public key. Only V_j's private key can decrypt these messages, ensuring data confidentiality and integrity. Similarly, vehicle V_j encrypts a message to V_i using similar methods. During trust score updates, vehicle V_i needs to compute $G_i = f_i P$ and $G_i^* = f_i P_i$. Subsequently, TAI_i computes $G_i^* = S_i G_i$, and authenticates the identity of the vehicle user using $RIW_i ? = F_i \oplus G_i^* \oplus A_i$.

(4) *Resistance to data replay attack*: Timestamps are employed in interactions between entities. For instance, upon receiving a message, TAI_i checks the verification of $\Delta CT < |CT_r - CT_i|$, and confirms that the generated random numbers are unique and unpredictable. This design is intended to enhance the system security and minimize the risk of data replay attacks.

(5) *Timed heartbeat mechanism and pull services*: TAI_i periodically requests a list of reliable CSS_i to ensure the system's overall security. This prevents CSS_i from crashing due to excessive data requests, which would otherwise hinder supervision of vehicle behaviors. Additionally, CSS_i regularly sends available information to TAI_i optimizing service utilization through a load balancing algorithm.

6 Performance Evaluation

6.1 Comparison of Computation Overhead

In this subsection, we compare our proposed scheme with three alternative schemes to show its superior performance. Due to resource constraints, we conduct our tests using a personal computer (Lenovo, with an Intel(R) Core(TM) i9-12900H@ 2.50 GHz processor, 16 GB main memory, and the windows11 operation system) to serve as the platforms for the testing of three entities: Vehicle, TAI and CSS. Utilizing the renowned MIRACL library, we execute 5000 iterations of the operations and to obtain the average time required for each. Results are presented in Table 2. The Montgomery curve scalar multiplication (T_{msm}) is computationally faster than the Weierstrass curve scalar multiplication (T_{wsm}). Consequently, we adopt the same scheme used in SMAKA [44]. The execution time of Exclusive OR and concatenation operations are negligible, and thus not addressed in this paper. Additionally, we present a comparative analysis of the computational overhead of our scheme and other schemes in different scenarios in Tables 3–5. As other existing schemes [5,44,45] closely resemble our proposed architecture, we will solely focus on discussing our proposed scheme in the following subsections.

Table 2: Several cryptographic operation symbol definitions

Symbol	Description	Time (ms)
T_{uh}	Unidirectional hash arithmetic	0.0012
T_{wsm}	Weierstrass curve scale multiplication in ECC	0.4190
T_{msm}	Montgomery curve scale multiplication in ECC	0.1790
T_{sed}	Symmetric encryption/decryption	0.0212
T_{pao}	Point addition operation in ECC	0.0062
T_{cpo}	Chebyshev polynomial operation	0.0328
T_{mho}	MapToPoint hash operation	0.2384

Table 3: Comparison of computational overhead evaluation of different scenarios in Case I

	Vehicle	TAI/TA/RA	CSS/CSP/UVA
ECPP [5]	$3T_{wsm} + 8T_{uh} =$ 1.2666 (ms)	$2T_{wsm} + 10T_{uh} =$ 0.85 (ms)	$3T_{wsm} + 7T_{uh} =$ 1.2654 (ms)
SMAKA [44]	$3T_{msm} + 9T_{uh} + T_{sed} =$ 0.569 (ms)	$2T_{msm} + 10T_{uh} + T_{sed} =$ 0.3912 (ms)	$3T_{msm} + 7T_{uh} =$ 0.5454 (ms)
UAV-assisted [45]	$7T_{mho} + 3T_{cpo} =$ 1.7672 (ms)	$6T_{mho} + T_{cpo} =$ 1.4632 (ms)	$10T_{mho} + 4T_{cpo} =$ 2.5152 (ms)
Our	$2T_{msm} + 5T_{uh} =$ 0.364 (ms)	$2T_{msm} + 2T_{uh} =$ 0.3604 (ms)	$T_{msm} + 2T_{uh} =$ 0.1814 (ms)

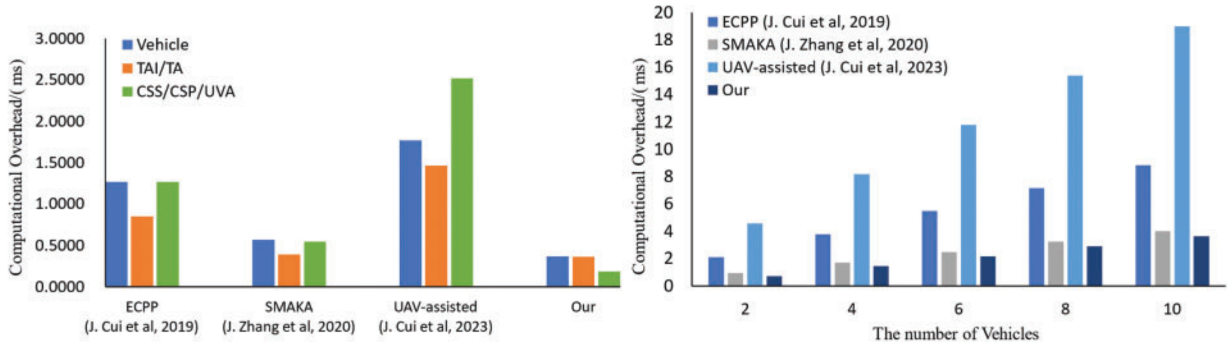
Table 4: Comparison of computational overhead evaluation of different scenarios in Case II

	n Vehicle	TAI/TA/RA	CSS/CSP/UVA
ECPP [5]	$(1 + 2n)\ T_{wsm} + (5 + 3n)\ T_{uh} = 0.8416n + 0.425$ (ms)	$2nT_{wsm} + 10nT_{uh} = 0.85n$ (ms)	$nT_{wsm} + 7nT_{uh} = 0.4274n$ (ms)
SMAKA [44]	$(1 + 2n)\ T_{msm} + (5 + 4n)\ T_{uh} + nT_{sed} = 0.384n + 0.185$ (ms)	$(1 + n)\ T_{msm} + (2 + 8n)\ T_{uh} + nT_{sed} = 0.2098n + 0.1814$ (ms)	$(2 + n)\ T_{msm} + (2 + 5n)\ T_{uh} = 0.185n + 0.3604$ (ms)
UAV-assisted [45]	$(7n + 4)\ T_{mho} + (4n + 1)\ T_{cpo} = 1.8n + 0.9864$ (ms)	$(6 + 7n)\ T_{mho} + (1 + n)\ T_{cpo} = 1.7016n + 1.4632$ (ms)	$10nT_{mho} + 4nT_{cpo} = 2.5152n$ (ms)
Our	$2nT_{msm} + (4n + 1)\ T_{uh} = 0.3628n + 0.0012$ (ms)	$(n + 1)\ T_{msm} + 2nT_{uh} = 0.1814n + 0.179$ (ms)	$nT_{msm} + 2nT_{uh} = 0.1814n$ (ms)

Table 5: Comparison of computational overhead evaluation of different scenarios in Case III

	n Vehicle	TAI/TA/RA	m CSS/CSP/UVA
ECPP [5]	$(m+2)\,nT_{wsm} + (5m+3)\,nT_{uh} = 0.425mn + 0.8416n$ (ms)	$(mn+n)\,T_{wsm} + (3n+7mn)\,T_{uh} = 0.4274mn + 0.4226n$ (ms)	$3mnT_{wsm} + 7mnT_{uh} = 1.2654mn$ (ms)
SMAKA [44]	$(m+2)\,nT_{msm} + (5m+4)\,nT_{uh} + nT_{sed} = 0.185mn + 0.384n$ (ms)	$(m+n)\,T_{msm} + (2m+2n+6mn)\,T_{uh} + nT_{sed} = 0.0072mn + 0.1814m + 0.2026n$ (ms)	$(2m+mn)\,T_{msm} + (2m+5mn)\,T_{uh} = 0.3604m + 0.185mn$ (ms)
UAV-assisted [45]	$(7+4m)\,nT_{mho} + (4+m)\,nT_{cpo} = 0.9864mn + 1.8n$ (ms)	$(6+7n)\,mT_{mho} + (1+n)\,mT_{cpo} = 1.4632m + 1.7016mn$ (ms)	$(3+7m)\,nT_{mho} + 4mnT_{cpo} = 0.7152n + 1.8mn$ (ms)
Our	$2nT_{msm} + (4+m)\,nT_{uh} = 0.0012mn + 0.3628n$ (ms)	$(n+m)\,T_{msm} + 2mnT_{uh} = 0.0024mn + 0.179m + 0.179n$ (ms)	$mnT_{msm} + 2mnT_{uh} = 0.1814mn$ (ms)

(1) *Case I*: In this case, a single vehicle interacts solely with a CSS node system. Table 3 illustrates a comparison of the computational overhead between our scheme and three other schemes [5,44,45] across the login, authentication, and key negotiation phases. These compared schemes also adopt a three-layer architecture similar to our scheme. During the login phase in our scheme, the vehicle needs to compute two one-way hash function operations. The authentication and key negotiation phases do not necessitate additional operations. Thus, the computational overhead for the vehicle is $2T_{msm} + 5T_{uh} = 0.364$ ms. When a vehicle user sends a message to the TAI, the TAI needs to perform two scalar multiplication and a one-way hash function operation, thus its computational overhead is $2T_{msm} + 2T_{uh} = 0.3604$ ms. Simultaneously, upon receiving the message from the TAI, the CSS undertakes one scalar multiplication and two one-way hash operations, with a computational overhead of $T_{msm} + 2T_{uh} = 0.1814$ ms. To illustrate the superiority of our scheme in computational overhead compared to other schemes, The results of the computational cost comparison for a vehicle to a CSS are shown in Fig. 7a. It is evident from the figure that our scheme surpasses scheme [5] and scheme [44] significantly. Specifically, scheme [45] exhibits notably higher computational costs compared to our scheme.



(a): Case I

(b): Case II

Figure 7: Computational overhead (a Vehicle to a CSS; n Vehicle to a CSS, n from 2 to 10) [5,44,45]

(2) *Case II*: In this case, a single CSS system is interacted with by n vehicles. Table 4 presents the computational costs incurred by these n vehicles, spanning from successful login to message transmission to the CSS, allowing for comparison with the literature [5,44,45]. Specifically, vehicles must execute $2n$ Montgomery curve multiplications and $4n + 1$ one-way irreversible hash operations, resulting in a computational cost of $2nT_{msm} + (4n + 1)\ T_{uh} = 0.3628n + 0.0012$ ms. Furthermore, the computational expenses for TAI and CSS amount to $(n + 1)T_{msm} + 2nT_{uh} = 0.1814n + 0.179$ ms and $nT_{msm} + 2nT_{uh} = 0.1814n$ ms, respectively. To demonstrate that our proposed scheme is more efficient in terms of computational costs compared to other schemes, Fig. 7b presents a comparative analysis of the computational costs for n vehicles interacting with a single CSS. To facilitate the representation, the value of n ranges from 2 to 10. The results clearly indicate that our scheme significantly outperforms other approaches.

(3) *Case III*: In this case, n vehicles transmit authentication messages to m CSSs. Table 5 displays the computational overheads calculated by our scheme and the comparison schemes [5,44,45] during the login, authentication message transmission to TAI and key negotiation phases, and post-receipt of messages from TAI by the CSSs. Given our emphasis on the computational overhead of resource-constrained vehicles in practical settings, we solely present the computational overhead of vehicle entities. In our scheme, n vehicles logging in and sending messages to m CSSs requires computing $2n$ Montgomery curve proportional multiplications and $(4 + m)n$ hashes, so the computational overhead is: $2nT_{msm} + (4 + m)\ nT_{uh} = 0.0012mn + 0.3628n$ ms. When TAI receives the authentication and request messages sent from the vehicle and CSS, it needs to compute $(n + m)$ times the Montgomery curve scaling multiplication, and $2mn$ hashes, so the computational overhead is $(n + 1)T_{msm} + 2nT_{uh} = 0.1814n + 0.179$ ms. After CSS receives the request sent by TAI, it needs to compute mn times the Montgomery curve scaling multiplication and $2mn$ times the hash. The computational overhead of CSS is therefore $mnT_{msm} + 2mnT_{uh} = 0.1814mn$ ms. To visually demonstrate the advantages of our scheme, Fig. 8 presents a comparison between our scheme and others. In this scenario, the number of vehicles ranges from 2 to 10, while the number of m CSSs is fixed at 10 for consistency. The figure illustrates that with multiple CSS, the computational load of this scheme is lower than that of scheme [5] and scheme [44] as n increases. Notably, the benefits of this scheme over scheme [45] become more evident with a higher number of vehicles.

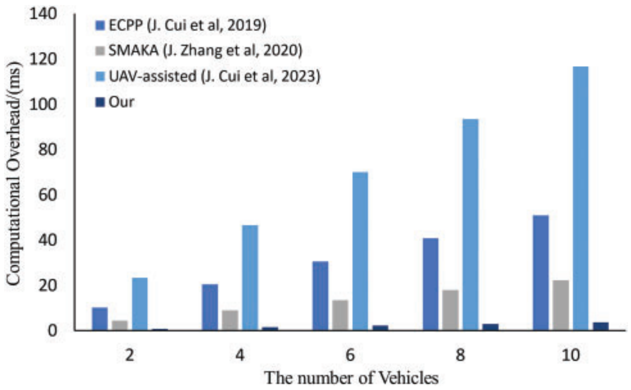


Figure 8: Computational overhead (n Vehicle to m CSS, n from 2 to 10, $m = 10$) [5,44,45]

The analysis above demonstrates that, our proposed scheme has a lower total computational overhead than the schemes [5,44,45], exhibiting a significant advantage. This reduction in computational overhead enhances the efficiency of the trust evaluation mechanism, making it more suitable for real-time applications in vehicular networks where vehicles have limited computational resources.

6.2 *Comparison of Communication Overhead*

Initially, we establish an identical security condition, where $P_1 = 20$ B, $p = 64$ B, the size of G_1 is $20 \times 2 = 40$ B, and G_2 is $64 \times 2 = 128$ B. Additionally, the hash output is 20 B in size, the encryption and decryption output is 16 B, and the size for both the timestamp and request information is 4 B. Elliptic curve proportional multiplication belong to G_1, yields an output of 40 B. Tables 6–8 display the communication overhead during the authentication and key negotiation stages for the aforementioned cases, respectively. As the AKA stage of the schemes [5,44,45] is similar to our scheme, this paper introduces the proposed scheme.

Table 6: Evaluation of communication overhead for different scenarios in Case I

Scheme	Number of rounds for sending messages	Interaction flow between entities	Communication cost
ECPP [5]	6	$V_i \to TA \to CSP \to TA \to CSP \to V_i \to CSP$	$88 + 4 + 88 + 124 + 104 + 20 = 428$ (B)
SMAKA [44]	4	$CSP \to V_i \to RA \to CSP \to V_i$	$84 + 76 + 104 + 68 = 368$ (B)
UAV-assisted [45]	4	$V_i \to TA \to U_j \to TA \to V_i$	$100 + 84 + 104 + 104 = 392$ (B)
Our	4	$V_i \to TAI \to CSS \to TAI \to V_i$	$48 + 16 + 24 + 44 = 132$ (B)

Table 7: Evaluation of communication overhead for different scenarios in Case II

Scheme	Number of rounds for sending messages	Interaction flow between entities	Communication cost
ECPP [5]	$6n$	$V_i \to TA \to CSP \to TA \to CSP \to V_i \to CSP$	$88n + 4n + 88n + 124n + 104n + 20n = 428n$ (B)
SMAKA [44]	$4n$	$CSP \to V_i \to RA \to CSP \to V_i$	$84 + 76n + 104n + 68n = 248n + 84$ (B)
UAV-assisted [45]	$4n$	$V_i \to TA \to U_j \to TA \to V_i$	$100n + 84n + 104n + 104n = 392n$ (B)
Our	$4n$	$V_i \to TAI \to CSS \to TAI \to V_i$	$48n + 16n + 24n + 44n = 132n$ (B)

(1) *Case I*: Table 6 lists the rounds required for a vehicle to send information to the CSS node and the interaction among various entities, and calculates the communication overhead for each process. During the vehicle login, authentication, and key negotiation stages, the proposed scheme requires four rounds of information transmission. These four pieces of information are $D1 = \{M_i, CT_i, G_i\}$, $D2 = \{M_i, VID_i, CT_i, v\}$, $D3 = \{M_j, CT_i, r_i\}$, and $D4 = \left\{M_j, CT_i\right\}$. And the communication overheads of them are $|D1| = (4 + 4 + 40) = 4$ B, $|D2| = (4 + 4 + 4 + 4) = 16$ B, $|D3| = (20 + 4 + 20) = 44$ B,

$|D4| = (20 + 4) = 24$ B, respectively. Fig. 9 illustrates a comparative analysis between our scheme and alternative schemes for a single vehicle to a CSS system, aiming to present the scheme's advantages more clearly. It is evident from the figure that our scheme exhibits reduced communication overhead compared to schemes [5,44,45].

Table 8: Evaluation of communication overhead for different scenarios in Case III

Scheme	Number of rounds for sending messages	Interaction flow between entities	Communication cost
ECPP [5]	$5mn + n$	$V_i \to TA \to CSP \to TA \to CSP \to V_i \to CSP$	$88n + 4mn + 88\,mn + 124\,mn + 104\,mn + 20\,mn = 88n + 340mn$ (B)
SMAKA [44]	$2mn + m + n$	$CSP \to V_i \to RA \to CSP \to V_i$	$84m + 76n + 104mn + 68mn = 172mn + 84m + 76n$ (B)
UAV-assisted [45]	$3mn + n$	$V_i \to TA \to U_j \to TA \to V_i$	$100n + 84mn + 104mn + 104mn = 292mn + 100n$ (B)
Our	$3mn + n$	$V_i \to TAI \to CSS \to TAI \to V_i$	$48n + 16mn + 44mn + 24mn = 84mn + 48n$ (B)

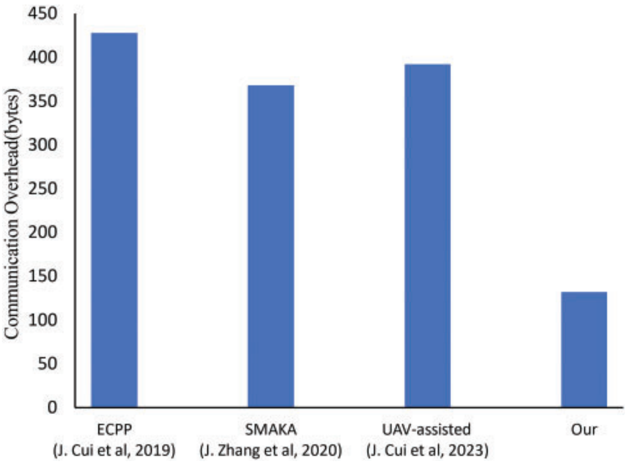
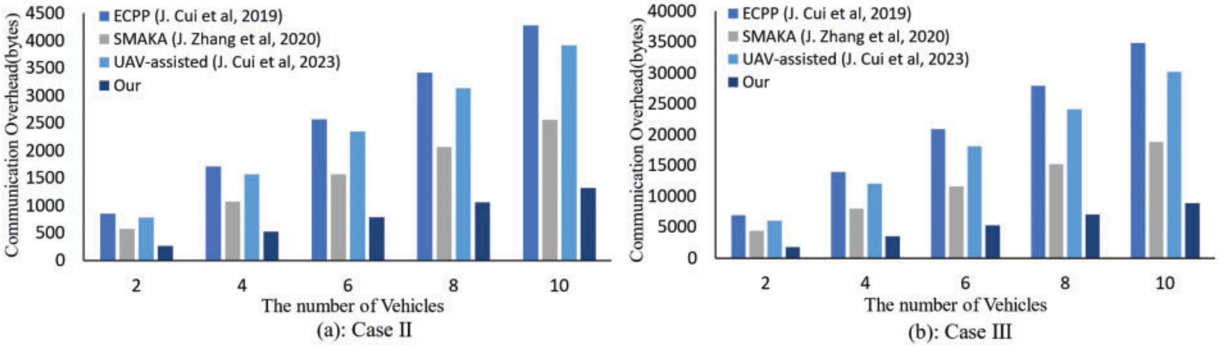


Figure 9: Evaluation of communication overhead for different scenarios (a Vehicle to a CSS) [5,44,45]

(2) *Case II*: As illustrated in Table 7, the proposed scheme necessitates $4n$ rounds for n vehicles corresponding to a CSS system in the login authentication and key negotiation phases. The communication overhead of the data transmitted in each round is detailed below: $n\,|D1| = n\,(4 + 4 + 40) = 48n$ B, $n\,|D2| = n\,(4 + 4 + 4 + 4) = 16n$ B, $n\,|D3| = n\,(20 + 4 + 20) = 44n$ B, $n\,|D4| = (20 + 4) = 24n$ B. So total communication overhead is:$n\,|D1| + n\,|D2| + n\,|D3| + n\,|D4| = 48n + 16n + 44n + 24n = 132n$ B. The table demonstrates that the communication overhead in this paper is significantly lower than in other schemes [5,44,45]. To further illustrate the advantages of our scheme, Fig. 10a provides a comparative analysis of the communication overhead for n vehicles interacting with a

single CSS, where n ranges from 2 to 10. The results clearly show that our scheme significantly reduces communication overhead compared to other schemes.



(a): Case II

(b): Case III

Figure 10: Communication overhead (n Vehicle to a CSS, n from 2 to 10; n Vehicle to m CSS, n from 2 to 10, $m = 10$) [5,44,45]

(3) *Case III*: As shown in Table 8, the proposed scheme requires ($3mn + n$) rounds for multiple vehicles corresponding to multiple CSSs during the login, authentication, and key negotiation phases. Since the vehicle sends the vehicle information and the TAI selects the appropriate CSS for service, only $n$vehicles corresponding to m CSSs are required each time. However, the schemes [5,44,45] incurs significant communication cost, which require $5mn + n$, $2mn + m + n$, and $3mn + n$ interactions, respectively. In this paper, the total communication overhead in this case is $n\,|D1| + mn\,|D2| + mn\,|D3| + n\,|D4| = 48n + 16mn + 44mn + 24mn = 84mn + 48n$ B. To demonstrate our advantage in comparison with other schemes regarding communication overhead, Fig. 10b illustrates the comparison of communication overhead in relation to a CSS from a varying number of vehicles to m CSSs. Since we emphasize the communication overhead of resource-constrained vehicles in practical applications, only the communication overhead of different numbers of vehicles is presented. For clarity, where m is set to 10 and n ranges from 2 to 10. From the figure, it is evident that as the number of vehicles increases with multiple m, this scheme exhibits lower communication costs compared to other schemes.

The above analysis, evidently show that the communication cost of our proposed scheme is lower than that of the related schemes [5,44,45]. This reduction in communication overhead enables the trust evaluation mechanism to handle more nodes and interactions with reduced delay, improving scalability.

6.3 Comparison of Storage Overhead

In system evaluation, storage overhead plays a crucial role. Given that the size of G_1 is 40 B and G_2 is 128 B, we assume that the real identity and password of each entity are 40 B, while the key, session key, master key, and private key are 128 B. Additionally, the group size is 128 B, with the generator of the group also 128 B, and the order of the generator 4 B. The public key encryption output is 40 B. Other parameter sizes are in the communication overhead section and will not be repeated here. Tables 9–11 provide a comparative analysis of the storage overhead of our proposed scheme and other schemes in various scenarios. Since the existing schemes [5,44,45] are structurally similar to ours, the following subsections will focus on the comparative advantages of our proposed scheme.

Table 9: Comparison of storage overhead evaluation of different scenarios in Case I

	Vehicle	TAI/TA/RA	CSS/CSP/UVA
ECPP [5]	$40 + 40 + 20 + 20 + 128 = 248$ (B)	$128+40+20+20+20 = 228$ (B)	$20 + 20 + 40 + 128 = 208$ (B)
SMAKA [44]	$40 + 40 + 20 + 20 + 128 + 128 + 20 + 20 + 4 = 420$ (B)	$20+20+4+20+20+4+20+128+20+128+4+128 + 20 + 20 = 556$ (B)	$20 + 128 + 4 + 128 + 20+20+4+20+20 = 364$ (B)
UAV-assisted [45]	$20 + 20 + 4 + 40 + 40 + 20 = 144$ (B)	$40 + 4 + 40 + 4 + 40 + 128 = 256$ (B)	$20+20+4+20+20 = 84$ (B)
Our	$40 + 40 + 20 + 40 + 40 + 20 + 20 = 220$ (B)	$40 + 128 + 20 + 20 + 4 = 212$ (B)	$20 + 40 + 4 = 64$ (B)

Table 10: Comparison of storage overhead evaluation of different scenarios in Case II

	n Vehicle	TAI/TA/RA	CSS/CSP/UVA
ECPP [5]	$40n + 40n + 20n + 20n + 128n = 248n$ (B)	$128n + 40n + 20n + 20n + 20n = 228n$ (B)	$20n + 20n + 40n + 128n = 208n$ (B)
SMAKA [44]	$40n + 40n + 20n + 20n + 128n + 128n + 20n + 20n + 4n = 420n$	$20n + 20n + 4n + 20n + 20n + 4n + 20n + 128n + 20n + 128n + 4n + 128n + 20n + 20n = 556n$ (B)	$20n + 128n + 4n + 128n + 20n + 20n + 4n + 20n + 20n = 364n$
UAV-assisted [45]	$20n + 20n + 4n + 40n + 40n + 20n = 144n$ (B)	$40n + 4n + 40n + 4n + 40n + 128n = 256n$ (B)	$20n + 20n + 4n + 20n + 20n = 84n$ (B)
Our	$40n + 40n + 20n + 20n + 20n + 40n + 40n = 220n$ (B)	$40n + 128n + 20n + 20n + 4n = 212n$ (B)	$20n + 40n + 4n = 64n$ (B)

Table 11: Comparison of storage overhead evaluation of different scenarios in Case III

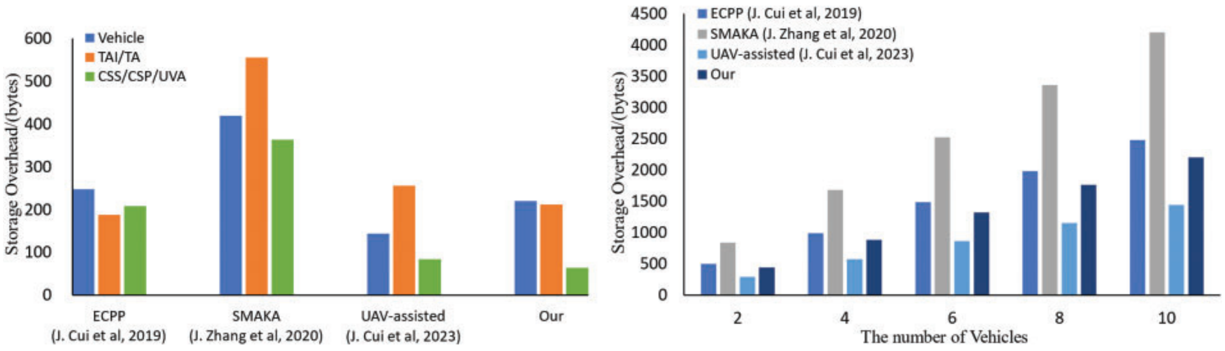
	n Vehicle	TAI/TA/RA	m CSS/CSP/UVA
ECPP [5]	$40n + 40n + 20mn + 20mn + 128mn = 208mn + 80n$ (B)	$128mn + 40mn + 20mn + 20mn + 20mn = 228mn$ (B)	$20n + 20mn + 40mn + 128mn = 188mn + 20n$ (B)
SMAKA [44]	$40n + 40n + 20mn + 20mn + 128mn + 128n + 20n + 20n + 4n = 168mn + 252n$ (B)	$20n + 20n + 4n + 20n + 20n + 4n + 20mn + 128mn + 20mn + 128mn+4mn+128mn+20mn+ 20mn = 468mn + 88n$ (B)	$20mn + 128mn + 4mn + 128mn+20mn+20mn+ 4mn + 20mn + 20mn = 364mn$ (B)
UAV-assisted [45]	$20mn + 20mn + 4mn + 40n + 40n + 20mn = 64mn + 80n$ (B)	$40mn + 4mn + 40mn + 4mn + 40mn + 128mn = 256mn$ (B)	$20mn + 20mn + 4mn + 20mn + 20mn = 84mn$ (B)

(Continued)

Table 11 (continued)

	n Vehicle	TAI/TA/RA	m CSS/CSP/UVA
Our	$40mn + 40mn + 20mn + 20mn + 20mn + 40n + 40n = 140mn + 80n$ (B)	$40mn + 128mn + 20mn + 20mn + 4mn = 212mn$ (B)	$20mn + 40mn + 4mn = 64mn$ (B)

(1) *Case I*: In this case, a single vehicle interacts with a single CSS node. As shown in Table 9, we compare the storage overhead of our scheme with other schemes during the registration, login, authentication, and key agreement phases. When a vehicle registers with the TAI, it needs to store two parameters encrypted with the TAI's public key. The parameters required for the login phase are already stored during registration, eliminating the need for additional storage at that stage. For key agreement between vehicles, each vehicle stores the unique identifier and the hashed value of the password for both itself and the interacting vehicle. Additionally, vehicles interacting with the TAI need to store two hash functions, H_1 and H, generated by the TAI during initialization for encryption computations. As a result, the total storage overhead for a single vehicle is $40+40+20+20+20+40+40 = 220$ B. For the TAI, the required storage includes its public-private key pair, two hash functions, and trust scores for vehicles during key negotiation, amounting to $40 + 128 + 20 + 20 + 4 = 212$ B. The CSS stores a hash function and the elliptic curve scalar multiplication for registration, as well as the request information sent by the vehicle during authentication and key agreement, resulting in a total storage overhead of $20 + 40 + 4 = 64$ B. Since the CSS continuously monitors traffic conditions in real time and is typically delegated to cameras, only the storage overhead of interaction information for trust evaluation is considered. To clearly demonstrate the advantages of our scheme, Fig. 11a provides a comparative analysis of the storage overhead for a single vehicle interacting with a single CSS. As shown in Table 9 and Fig. 11a, the storage overhead of our scheme demonstrates reasonable performance in resource-constrained scenarios. Compared to existing schemes [5,44,45], our proposed solution is suited for complex vehicular network environments.



(a): Case I

(b): Case II

Figure 11: Storage overhead (a Vehicle to a CSS; n Vehicle to a CSS, n from 2 to 10) [5,44,45]

(2) *Case II*: In this case, n vehicles interact with a single CSS system. Table 4 outlines the storage overhead incurred from the registration phase to the transmission of information to the CSS. Specifically, each of the n vehicles store two parameters encrypted with the TAI's public key, along

with the hash values generated during key agreement between vehicles. Each vehicle also needs to store its unique identifier, password, and the two hash functions generated by the TAI for secure data transmission, resulting in a total storage overhead of $40n + 40n + 20n + 20n + 20n + 40n + 40n = 220n$ B. The TAI must store the public-private key pairs for n instances, two hash functions, and trust score request information for n instances, leading to a storage overhead of $40n + 128n + 20n + 20n + 4n = 212n$ B. The CSS is required to store n instances of hash values, elliptic curve scalar multiplications, and the request information sent by the vehicles, amounting to a storage overhead of $20n + 40n + 4n = 64n$ B. To clearly illustrate the storage overhead of our scheme, Fig. 11b provides a comparative analysis focused on the communication overhead for resource-constrained vehicles in practical applications. For clarity, the number of vehicles is set between 2 and 10. As observed from Table 10 and Fig. 11b, the storage overhead of our scheme remains within manageable limits, demonstrating its practicality in resource-constrained vehicular networks.

(3) *Case III*: In this case, n vehicles transmit authentication messages to m CSSs. We compare the storage overhead of our proposed scheme with other schemes [5,44,45], focusing on different stages such as registration, login, key agreement, and inter-entity interaction, as shown in Table 11. Given the focus on resource-constrained vehicles in practical environments, we specifically discuss the computational and storage overhead for the vehicle entities. In our scheme, each vehicle is required to store mn instances of two public key encryption parameters and the hash values generated between vehicles. Additionally, vehicles must store their unique identifiers, passwords, and two hash functions generated by the TAI for secure data transmission. Furthermore, each vehicle needs to store n instances of its unique identifier and password. Therefore, the total storage overhead for each vehicle is $40mn + 40mn + 20mn + 20mn + 20mn + 40n + 40n = 140mn + 80n$ B. For the TAI, the storage overhead consists of mn instances of public-private key pairs, two hash functions, and vehicle request information. Hence, the TAI's total storage overhead is $40mn + 128mn + 20mn + 20mn + 4mn = 212mn$ B. Finally, for the CSS, the storage overhead involves mn instances of one-way hash values, elliptic curve scalar multiplications, and vehicle request information, resulting in the following storage requirement: $20mn + 40mn + 4mn = 64mn$ B. To visually illustrate the storage overhead, Fig. 12 compares our scheme with other schemes as the number of vehicles n increases from 2 to 10, with the number of CSS m fixed at 10 for consistency. The results show that as n increases, our scheme consistently demonstrates lower storage overhead compared to scheme [5] and scheme [44]. Compared to scheme [45], our scheme remains a suitable and effective choice for vehicular networks, particularly as the number of vehicles grows.

The above analysis demonstrates that our proposed scheme offers a significant advantage in terms of total storage overhead. Although the storage overhead on the vehicle side is higher compared to scheme [45], our scheme provides clear benefits over scheme [5] and scheme [44]. Additionally, in terms of storage overhead at the TAI and CSS sides, our scheme performs more efficiently than schemes [5,44,45]. By employing a trust evaluation mechanism, our scheme optimizes storage distribution by focusing on real-time monitoring and dynamic trust updates, reducing the need for excessive data retention. This enhances both the performance and adaptability of the scheme in vehicular networks, particularly in environments where efficient storage use is critical for ensuring security and maintaining trust during real-time communication.

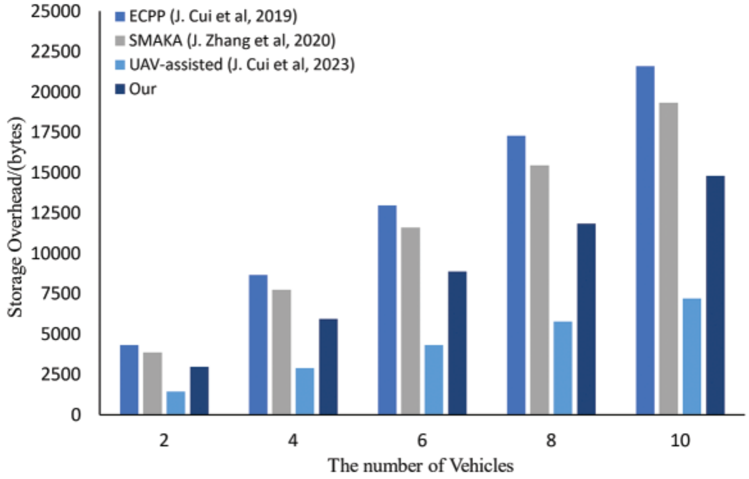


Figure 12: Storage overhead (n Vehicle to m CSS, n from 2 to 10, $m = 10$) [5,44,45]

7 Discussion

The proposed scheme employs a trust evaluation mechanism, where CSS plays a crucial role in real-time monitoring of road conditions and vehicle interactions. This real-time supervision is vital, as failure to detect malicious behavior can lead to serious security issues. In our scheme, we assume that the CSS is fully trusted. If malicious vehicles attempt to send deceptive information during vehicle-to-vehicle interactions, the reporting vehicle can file a complaint to the CSS via the TAI. Given that the CSS continuously monitors road conditions, both the behavior of the reporting and the reported vehicles will be accurately assessed, minimizing the risks of false positives or negatives. The CSS then applies trust evaluation to reward or penalize the involved vehicles accordingly. This mechanism is essential, as real-time monitoring and punishing malicious vehicles in practical vehicular networks would otherwise entail significant computational, communication, and storage overhead. By utilizing this trust-based approach, our scheme effectively mitigates these challenges, thus fostering a more secure vehicular network environment.

However, in practical scenarios, the assumption of a fully trusted CSS may be limiting, as achieving complete trustworthiness in real-world deployments is challenging. Additionally, vehicular networks are inherently dynamic systems. In our scheme, different regions are managed by distinct RSUs, and interactions between vehicles in different regions are facilitated through cluster head vehicles. Vehicles entering a new region do not have access to the previous interaction data of that region, while vehicles exiting the region no longer have access to subsequent interactions occurring within it. Failure to maintain this boundary could introduce security vulnerabilities. This dynamic nature poses a challenge to the scalability of the scheme, as vehicles frequently enter and exit different regions.

To address these challenges, integrating federated learning with blockchain technology offers a promising solution. Federated learning allows vehicles to collaboratively learn trust models without sharing sensitive data, preserving privacy while improving the accuracy of trust evaluations. Meanwhile, blockchain ensures the integrity and immutability of trust records across different regions, allowing vehicles entering new areas to verify past interactions without accessing the full historical data. This combination enhances the scalability of the scheme by securely managing dynamic vehicle interactions and mitigating trust issues without overloading the network with communication and computational overhead. By leveraging these technologies, our scheme could more effectively handle the inherent dynamism of vehicular networks while maintaining security and efficiency.

8 Conclusion

This paper proposed a distributed network architecture based on trust scores. It aimed to improve the vehicular network environment by enabling numerous vehicles to report malicious ones disseminating erroneous information to the TAIs. Specifically, vehicles sent reporting information to the TAIs, which then relayed this information to the CSSs for detection. Based on the CSSs' s detection results, the TAI increased or decreased the trust scores of the reporting and the reported vehicles, thus enhancing the security across the vehicular network. Regarding security, we demonstrated the proposed scheme achieved the security objectives. Compared to other schemes, our scheme had lower computation and communication overhead and exhibited superior scalability. In the future, we will further improve this scheme by using a federated learning-based trust score evaluation mechanism to accommodate more complex scenarios.

Acknowledgement: The authors would like to express our sincere gratitude and appreciation to each other for our combined efforts and contributions throughout the course of this research paper. Thanks to the reviewers and experts of your magazine for their valuable opinions on the article revision.

Funding Statement: This work is supported the by Anhui Provincial Natural Science Foundation under Grant 2308085MF223; in part by the Open Fund of State Key Laboratory for Novel Software Technology under Grant KFKT2022B33; in part by the by the Foundation of Yunnan Key Laboratory of Service Computing under Grant YNSC23106; and in part by the Key Project on Anhui Provincial Natural Science Study by Colleges and Universities under Grant 2023AH050495, 2024AH051078 and Grant KJ2020A0513.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Wenming Wang, Zhiquan Liu, Guijiang Liu; data collection: Zhiquan Liu, Shumin Zhang; analysis and interpretation of results: Wenming Wang, Zhiquan Liu, Guijiang Liu; draft manuscript preparation: Zhiquan Liu, Shumin Zhang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] X. Ge, Z. Li, and S. Li, "5G software defined vehicular networks," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 87–93, Jul. 2017. doi: 10.1109/MCOM.2017.1601144.

[2] S. Bojjagani, Y. P. Reddy, T. Anuradha, P. V. Rao, B. R. Reddy and M. K. Khan, "Secure authentication and key management protocol for deployment of internet of vehicles (IoV) concerning intelligent transport systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 24698–24713, Sep. 2022. doi: 10.1109/TITS.2022.3207593.

[3] L. Song, G. Sun, H. Yu, X. Du, and M. Guizani, “FBIA: A fog-based identity authentication scheme for privacy preservation in internet of vehicles,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5403–5415, Mar. 2020. doi: 10.1109/TVT.2020.2977829.

[4] L. Yang, K. Yu, S. X. Yang, C. Chakraborty, Y. Lu and T. Guo, “An intelligent trust cloud management method for secure clustering in 5G enabled internet of medical things,” *IEEE Trans. Ind. Inf.*, vol. 18, no. 12, pp. 8864–8875, Nov. 2021. doi: 10.1109/TII.2021.3128954.

[5] 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 Trans. Inf. Forensics Secur.*, vol. 15, pp. 1654–1667, Oct. 2019. doi: 10.1109/TIFS.2019.2946933.

[6] W. Pi, P. Yang, D. Duan, C. Chen, and H. Li, “Malicious user detection for cooperative mobility tracking in autonomous driving,” *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4922–4936, Feb. 2020. doi: 10.1109/JIOT.2020.2973661.

[7] Q. Ding, J. Wang, X. Zhang, and D. K. Sung, “Modeling and characterization of the detection and suppression of bogus messages in vehicular ad hoc networks,” *IEEE Trans. Mob. Comput.*, vol. 22, no. 10, pp. 6027– 6040, Jun. 2022. doi: 10.1109/TMC.2022.3182005.

[8] C. Li *et al.*, “Federated hierarchical trust-based interaction scheme for cross-domain industrial IoT,” *IEEE Internet Things J.*, vol. 10, no. 1, pp. 447–457, Aug. 2022. doi: 10.1109/JIOT.2022.3200854.

[9] M. Gerlach, “Trust for vehicular applications,” in *Proc. 8th Int. Symp. Auton. Decentralized Syst.*, Sedona, AZ, USA, Jun. 2007, pp. 295–304.

[10] U. F. Minhas, J. Zhang, T. Tran, and R. Cohen, “Towards expanded trust management for agents in vehicular ad-hoc networks,” *Int. J. Comput. Intell. Theory Pract.*, vol. 5, no. 1, pp. 3–15, Jun. 2010.

[11] M. Raya, P. Papadimitratos, V. D. Gligor, and J. -P. Hubaux, “On data-centric trust establishment in ephemeral ad hoc networks,” in *Proc. IEEE INFOCOM 27th Conf. Comput. Commun.*, Apr. 2008, pp. 1238–1246.

[12] F. Dötzer, L. Fischer, and P. Magiera, “VARS: A vehicle ad-hoc network reputation system,” in *Proc. 6th IEEE Int. Symp. World Wireless Mobile Multimedia Netw.*, Jun. 2005, pp. 454–456.

[13] B. Mokhtar and M. Azab, “Survey on security issues in vehicular ad hoc networks,” *Alexandria Eng. J.*, vol. 54, no. 4, pp. 1115–1126, Dec. 2015. doi: 10.1016/j.aej.2015.07.011.

[14] Z. Tian, X. Gao, S. Su, and J. Qiu, “Vcash: A novel reputation framework for identifying denial of traffic service in internet of connected vehicles,” *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3901–3909, May 2019. doi: 10.1109/JIOT.2019.2951620.

[15] L. Sun, Q. Yang, X. Chen, and Z. Chen, “RC-chain: Reputation-based crowdsourcing blockchain for vehicular networks,” *J. Network Comput. Appl.*, vol. 176, Feb. 2021, Art. no. 102956. doi: 10.1016/j.jnca.2020.102956.

[16] A. Hbaieb, S. Ayed, and L. Chaari, “A survey of trust management in the Internet of Vehicles,” *Comput. Networks*, vol. 203, Feb. 2022, Art. no. 108558. doi: 10.1016/j.comnet.2021.108558.

[17] X. Huang, D. Ye, R. Yu, and L. Shu, “Securing parked vehicle assisted fog computing with blockchain and optimal smart contract design,” *IEEE/CAA J. Autom. Sin.*, vol. 7, no. 2, pp. 426–441, Mar. 2020. doi: 10.1109/JAS.2020.1003039.

[18] J. Cui, F. Ouyang, Z. Ying, L. Wei, and H. Zhong, “Secure and efficient data sharing among vehicles based on consortium blockchain,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8857–8867, Jun. 2021. doi: 10.1109/TITS.2021.3086976.

[19] J. Kang *et al.*, “Blockchain for secure and efficient data sharing in vehicular edge computing and networks,” *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4660–4670, Oct. 2018. doi: 10.1109/JIOT.2018.2875542.

[20] J. -M. Chen, T. -T. Li, and J. Panneerselvam, “TMEC: A trust management based on evidence combination on attack-resistant and collaborative internet of vehicles,” *IEEE Access*, vol. 7, pp. 148913–148922, Nov. 2018. doi: 10.1109/ACCESS.2018.2876153.

[21] U. Javaid, M. N. Aman, and B. Sikdar, “A scalable protocol for driving trust management in internet of vehicles with blockchain,” *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11815–11829, Jun. 2020. doi: 10.1109/JIOT.2020.3002711.

[22] H. Zhang, J. Liu, H. Zhao, P. Wang, and N. Kato, "Blockchain-based trust management for internet of vehicles," *IEEE Trans. Emerging Top. Comput.*, vol. 9, no. 3, pp. 1397–1409, Nov. 2020. doi: 10.1109/TETC.2020.3033532.

[23] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1495–1505, May 2018. doi: 10.1109/JIOT.2018.2836144.

[24] H. El-Sayed, H. Alexander, P. Kulkarni, M. A. Khan, R. M. Noor and Z. Trabelsi, "A novel multifaceted trust management framework for vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 20084–20097, Nov. 2022. doi: 10.1109/TITS.2022.3187788.

[25] J. Tang, X. Lu, Y. Xiang, C. Shi, and J. Gu, "Blockchain search engine: Its current research status and future prospect in Internet of Things network," *Future Gen. Comput. Syst.*, vol. 138, no. 11, pp. 120–141, Jan. 2023. doi: 10.1016/j.future.2022.08.008.

[26] M. Wazid, A. K. Das, and S. Shetty, "TACAS-IoT: Trust aggregation certificate-based authentication scheme for edge-enabled IoT systems," *IEEE Internet Things J.*, vol. 9, no. 22, pp. 22643–22656, Jun. 2022. doi: 10.1109/JIOT.2022.3181610.

[27] F. Qiao, J. Wu, J. Li, A. K. Bashir, S. Mumtaz and U. Tariq, "Trustworthy edge storage orchestration in intelligent transportation systems using reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4443–4456, Jun. 2020. doi: 10.1109/TITS.2020.3003211.

[28] S. Abdelhamid, H. S. Hassanein, and G. Takahara, "Vehicle as a resource (VaaR)," *IEEE Netw.*, vol. 29, no. 1, pp. 12–17, Jan. 2015. doi: 10.1109/MNET.2015.7018198.

[29] M. Ren, J. Zhang, L. Khoukhi, H. Labiod, and V. Vèque, "A unified framework of clustering approach in vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1401–1414, Aug. 2017. doi: 10.1109/TITS.2017.2727226.

[30] K. Zhang, J. Wang, C. Jiang, T. Q. Quek, and Y. Ren, "Content aided clustering and cluster head selection algorithms in vehicular networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, San Francisco, CA, USA, Mar. 2017, pp. 1–6.

[31] D. Zhang, H. Ge, T. Zhang, Y. -Y. Cui, X. Liu and G. Mao, "New multi-hop clustering algorithm for vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 4, pp. 1517–1530, Aug. 2018. doi: 10.1109/TITS.2018.2853165.

[32] M. Z. Alam, F. S. Abkenar, I. Adhicandra, S. Murali, and A. Jamalipour, "Low-delay path selection for cluster-based buffer-aided vehicular communications," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 9356–9363, Mar. 2020. doi: 10.1109/TVT.2020.2976926.

[33] W. Wang *et al.*, "Vehicle trajectory clustering based on dynamic representation learning of internet of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3567–3576, Jun. 2021. doi: 10.1109/TITS.2020.2995856.

[34] H. Zhong, L. Wang, J. Cui, J. Zhang, and I. Bolodurina, "Secure edge computing-assisted video reporting service in 5G-enabled vehicular networks," *IEEE Trans. Inform. Forensic Secur.*, vol. 18, pp. 3774–3786, Jun. 2023. doi: 10.1109/TIFS.2023.3287731.

[35] U. V. Vazirani and V. V. Vazirani, "Efficient and secure pseudo-random number generation," in *Found. Comput. Sci.*, Singer Island, FL, USA, Oct. 1984, pp. 193–202.

[36] R. Hou *et al.*, "Data forwarding scheme for vehicle tracking in named data networking," *IEEE Trans. Veh. Technol.*, vol. 70, no. 7, pp. 6684–6695, May 2021. doi: 10.1109/TVT.2021.3081448.

[37] Y. -C. Chu and N. -F. Huang, "An efficient traffic information forwarding solution for vehicle safety communications on highways," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 631–643, Dec. 2011. doi: 10.1109/TITS.2011.2177456.

[38] Spring, "Spring cloud," 2003. Accessed: Feb. 12, 2022. [Online]. Available: https://spring.io/projects/spring-cloud

[39] E. Chen, S. Wang, Y. Fan, Y. Zhu, and S. S. Yau, "SaaSC: Toward pay-as-you-go mode for software service transactions based on blockchain's smart legal contracts," *IEEE Trans. Serv. Comput.*, vol. 16, no. 5, pp. 3665–3681, Apr. 2023. doi: 10.1109/TSC.2023.3267489.

[40] M. Randles, D. Lamb, and A. Taleb-Bendiab, “A comparative study into distributed load balancing algorithms for cloud computing,” in *IEEE Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Perth, WA, Australia, Apr. 2010, pp. 551–556.

[41] T. Li, D. Baumberger, and S. Hahn, “Efficient and scalable multiprocessor fair scheduling using distributed weighted round-robin,” *ACM Sigplan Not.*, vol. 44, no. 4, pp. 65–74, Feb. 2009. doi: 10.1145/1594835.1504188.

[42] A. Singh, P. Goyal, and S. Batra, “An optimized round robin scheduling algorithm for CPU scheduling,” *Int. J. Comput. Sci. Eng.*, vol. 2, no. 7, pp. 2383–2385, Feb. 2010. doi: 10.48550/arXiv.1605.00362.

[43] L. Zhu, J. Cui, and G. Xiong, “Improved dynamic load balancing algorithm based on Least-Connection Scheduling,” in *IEEE Inf. Technol. and Mechatronics Eng. Conf.*, Chongqing, China, Dec. 2018, pp. 1858–1862.

[44] J. Zhang, H. Zhong, J. Cui, Y. Xu, and L. Liu, “SMAKA: Secure many-to-many authentication and key agreement scheme for vehicular networks,” *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 1810–1824, Dec. 2020. doi: 10.1109/TIFS.2020.3044855.

[45] J. Cui *et al.*, “A practical and provably secure authentication and key agreement scheme for UAV-assisted VANETs for emergency rescue,” *IEEE Trans. Network Sci. Eng.*, vol. 11, no. 2, pp. 1454–1468, Oct. 2023. doi: 10.1109/TNSE.2023.3323972.