Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 3646960, 20 pages
https://doi.org/10.1155/2022/3646960

Research Article

CLASRM: A Lightweight and Secure Certificateless Aggregate Signature Scheme with Revocation Mechanism for 5G-Enabled Vehicular Networks

Zhihua Wang,[1,2] **Haofan Wang**,[2] **Yongjian Wang**,[3] **and Xiaolong Yang**[1]

[1]*School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China*
[2]*School of Cyber Science and Engineering, Zhengzhou University, Zhengzhou 450001, China*
[3]*Lab of National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China*

Correspondence should be addressed to Yongjian Wang; wyj@cert.org.cn

Received 16 May 2021; Accepted 12 March 2022; Published 12 April 2022

Academic Editor: Antonio De Domenico

The rapid deployment of 5G technology has further strengthened the large-scale interconnection between sensing devices and systems and promoted the rapid development of smart cities and intelligent transportation systems. 5G-enabled vehicular networks take advantage of cellular vehicle-to-everything (C-V2X) technology to achieve the connection between moving vehicles, between vehicles and infrastructure, and between vehicles and the cloud, which can reduce the possibility of traffic jams and accidents, improve transportation efficiency, and realize automatic driving. Besides, 5G-enabled vehicular networks also provide infotainment services and industry application services. High-strength data transmission, however, will bring a serious burden of resource overhead, and there are hidden dangers of security and privacy in the communication process of vehicular networks. Some current vehicular network authentication schemes adopt public key infrastructure-based (PKI-based) and identity-based authentication methods to achieve conditional privacy preservation. Still, these schemes are too expensive and cannot address the problems of costly certificate management or risky key escrow. Some schemes use computationally complex bilinear pairing operations that result in low efficiency and do not consider the revocation of malicious nodes so that they cannot effectively prevent further malicious attacks. This paper proposes a lightweight certificateless aggregate signature (CLAS) scheme with a revocation mechanism suitable for 5G-enabled vehicular networks in response to the above problems. Our proposed scheme uses aggregation signature technology to aggregate multiple signatures into a single short signature, thus reducing communication overhead and storage overhead of road side units (RSUs). Furthermore, our proposed scheme utilizes the elliptic curve cryptography (ECC) to reduce verification time and computational overhead. Moreover, in order to prevent malicious users from sending invalid signatures to attack, our proposed scheme uses binary search to identify invalid signatures and introduces a cuckoo filter to revoke malicious users to prevent reattack. Finally, formal proof and experimental analysis show that our proposed scheme has greater advantages with respect to security and efficiency compared with the previous schemes.

1. Introduction

The large-scale commercial deployment of 5G technology has brought richer application scenarios, promoted the rapid development of smart cities and intelligent transportation systems, and provided more convenience for our lives and work. As an emerging communication technology, 5G enables higher data transmission rate and lower latency and supports direct communication between device-to-device (D2D). Simultaneously, as a crucial part of the smart city sensing system, the rapid development of intelligent sensing technology also makes innovative applications based on the Internet of Vehicles (IoVs) emerged in an endless stream. The traditional vehicular ad hoc networks (VANETs) primarily make use of the dedicated short range communication (DSRC) standard for vehicle-to-vehicle (V2V) communication and vehicle-to-infrastructure (V2I) communication. However, some studies have shown that the DSRC standard

has a high probability of collision [1–3], especially at higher vehicle density. Moreover, the DSRC standard also has defects in scalability, mobility, and latency, so that it cannot be well suited for delay-sensitive vehicle-to-everything (V2X) applications. Intelligent transportation systems and autonomous driving technologies put forward more stringent requirements on system performance such as communication rate, delay, and reliability of the IoVs. Therefore, C-V2X based on 5G communication technology that can provide low-latency and high-reliability V2X communication capabilities is proposed [4]. Multiple vehicles with intelligent sensing devices make use of C-V2X communication technology to jointly build 5G-enabled vehicular networks to provide a variety of V2X services and applications. For example, vehicles equipped with on-board units (OBUs) can communicate with adjacent vehicles and can also communicate with the RSUs to share information related to road safety and industrial applications. Then, vehicles and RSUs use the emergency and beacon information to make decisions in a timely manner to achieve intelligent driving assistance and reduce the possibility of traffic jams and accidents. In addition, vehicles can also collect sensing information in real time and send it to the cloud to provide services such as environmental monitoring and accident reporting, which realize vehicle-to-network (V2N) communication. Therefore, in order to provide high-quality applications and services, it is necessary to ensure the security and reliable connection of the IoVs.

In 5G-enabled vehicular networks, due to the openness and vulnerability of wireless channels, attackers can intercept, forge, and modify information through malicious means and even inject false information to cause vehicles to change lane or accelerate, which will cause unpredictable consequences [5]. Therefore, in 5G-enabled vehicular networks, the primary problem is how to ensure security of message transmission and exchange. It is not only necessary to authenticate identity of each message sender but also to check the integrity of the message, etc. Secondly, attackers can also obtain sensitive information such as the vehicles' trajectory through analysing the information sent by vehicles and then carry out some criminal acts, thereby reducing the enthusiasm of vehicles to join the IoVs [6]. Therefore, privacy leakage of vehicles is also a problem that must be solved. A pseudonym can be used to replace the vehicle's real identity for communication to ensure privacy and unlinkability. Moreover, a trusted third party that stores the vehicle's real identity is also needed to trace the anonymous vehicle and revoke its identity. Some conditional privacy-preserving authentication schemes have been surveyed [7–10]. Among them, two authentication schemes based on PKI have been proposed [7, 8] to meet security and privacy requirements. However, due to the large number of certificates that needs to be stored and the limited storage capacity of vehicles or RSUs, it is challenging to meet the requirements of PKI-based authentication mechanisms. Besides, in order to alleviate the problem of public key certificate management in PKI-based authentication schemes, many identity-based authentication schemes for VANETs have been proposed [9, 10] to realize conditional privacy preservation. However, once the private key generated by the private key generation centre is leaked, the problem of risky key escrow will appear. Fortunately, the certificateless signature (CLS) scheme can solve the above two problems well while retaining the advantages of the identity-based authentication mechanism [11–13].

There are also many resource-constrained devices in 5G-enabled vehicular networks. Large-scale communication between vehicles will place a severe burden on these devices. Therefore, the communication, computing, storage, and other overhead in IoVs are also worthy of our attention. Using aggregate signature or batch verification can reduce overhead so as to improve authentication efficiency and overcome the delay-sensitive problem [14]. Some schemes using bilinear pairing operations have been proposed [15–17], but these schemes incur huge computational overhead and low efficiency of message signing and verification. Therefore, these schemes cannot meet the applications requirements of low latency in 5G-enabled vehicular networks. Recently, Cui et al. [18] proposed a message authentication framework based on reputation score for delay-sensitive 5G-enabled vehicular networks, in which vehicles with poor reputation value cannot communicate with other vehicles. Moreover, Cui et al. [18] proposed an authentication scheme based on ECC, which supports batch authentication to reduce computational overhead. Taha and Shen [19] proposed a vehicular clustering algorithm based on speed, location, and signal strength and proposed a group authentication scheme for 5G scenarios to reduce the computational overhead of handover authentication. The above two schemes are both efficient and feasible, but they do not solve the problems of how to reduce communication overhead and how to revoke malicious users.

Therefore, in 5G-enabled vehicular networks, in order to achieve efficient communication between large-scale vehicles and to meet the requirements of security and privacy preservation, this paper proposes a CLAS scheme with revocation mechanism for 5G-enabled vehicular networks. The main contributions of this paper are summarized as below.

(1) In order to reduce the computational overhead, we propose a CLAS scheme based on ECC without using computationally complex bilinear pairing operations and map to point hash functions. When many vehicles send a large number of messages with signatures to the RSUs, there is huge communication overhead and storage overhead. To solve this problem, we utilize the aggregate signature technology to aggregate multiple signatures into a single short signature and verify it once

(2) In order to prevent malicious vehicles from influencing the aggregation verification and interfering with the normal communication of vehicles by inserting invalid signatures into the aggregation signatures, we make use of binary search to identify invalid signatures and take advantage of a cuckoo filter to construct a revocation mechanism to prevent malicious vehicles from attacking again. In order to prevent information injection attacks that often appear in

5G-enabled vehicular networks, the method with a random vector is used to ensure security in the signature aggregation phase

(3) Formal proof and security analysis can prove that the proposed CLAS scheme can meet the security and privacy preservation requirements in efficient communication. In addition, experimental results indicate that the proposed CLAS scheme has a particular improvement in computing and communication efficiency compared with the previous schemes

The organization of this paper is summarized as follows. We review related work in Section 2. We introduce background knowledge in Section 3. Section 4 describes the proposed CLAS scheme in detail. In Section 5, we give the formal proof and security requirements analysis of the proposed CLAS scheme. In Section 6, we compare our proposed scheme with other schemes in detail in terms of performance. Section 7 concludes this paper.

2. Related Work

This section will introduce the authentication schemes in traditional VANETs and the security protection methods applied to 5G-enabled vehicular networks, respectively.

At present, a variety of traditional PKI-based authentication schemes have been proposed [7, 8]. However, as the number of users increases, a large number of public key certificates will lead to a gradual increase in storage and communication overhead. In 1984, Shamir [20] proposed the identity-based public key cryptography (ID-PKC), which solved the problems existing in the use of certificates in traditional PKI-based schemes and reduced the overhead of certificate management. Although many identity-based authentication schemes have been proposed [20], it still cannot solve the key escrow problem. In these schemes, it is assumed that all users must completely trust the key generation centre (KGC), but this assumption is too ideal to be adopted practically in many applications.

Al-Riyami and Paterson [21] proposed an authentication scheme based on the certificateless public key cryptography (CL-PKC) in 2003, in which the user's private key is a combination of the partial private key generated by the KGC and the secret value generated by the user. After that, many CLS schemes and security models based on CL-PKC have been proposed [22–24]. Huang et al. [24] proved that the CLS scheme proposed in [21] could not resist the public key replacement attack and further proposed an improved CLS scheme. Subsequently, Yum and Lee [22] introduced a general CLS framework. However, the above solutions still have problems in terms of efficiency and are not suitable for the low-latency requirements of 5G-enabled vehicular networks.

Boneh et al. [14] first proposed the concept of aggregate signature in 2003. The aggregation signature technology can aggregate n signatures of n messages from n users into a single short signature to reduce the signature length and the authentication burden of RSUs. The validity of the aggregate signature is guaranteed by verifying the validity of each signature involved in the aggregate signature. Batch verification technology is also considered to be an effective method to improve verification efficiency. With this technique, multiple signatures from different signers of different messages can be verified at once. Subsequently, many CLAS schemes applied to VANETs have been proposed. In 2018, Cui et al. [25] proposed a CLAS scheme based on ECC, which has the advantages of certificateless public key cryptography and aggregate signature. This scheme can achieve trade-off between privacy preservation and traceability, and the security of the proposed scheme is proved by the random oracle model (ROM). Gayathri et al. [11] and Bayat et al. [12] proposed some effective CLS schemes for VANETs with batch verification. Unfortunately, Li and Zhang [13] found that the scheme proposed by Bayat et al. was insecure in their security model and proposed an improved scheme. In 2019, Kamil and Ogundoyin [26] proved that the CLS and CLAS schemes proposed by Cui et al. [25] were not secure against $Adv\,II$ adversary in the ROM and proposed an improved CLAS scheme for VANETs. Subsequently, Zhao et al. [27] proved that the scheme proposed by Kamil and Ogundoyin [26] could not resist $Adv\,I$ and $Adv\,II$ adversaries and proposed an improved CLAS scheme based on ECC. However, Thumbur et al. [28] pointed out that the construction of the CLS scheme [27] was incorrect in 2020. Zhong et al. [29] also proposed a privacy-preserving authentication scheme that realized full aggregation in VANETs, in which the length of the aggregate signature was kept constant, reducing communication and storage overhead. However, Kamil and Ogundoyin [30] proved that the scheme [29] was not secure in the standard security model by designing two specific attacks in 2020. Then, they modified the signature, verification, and aggregation verification algorithms [29] to make it more secure and effective. In 2020, Ren et al. [15] and Ali et al. [16], respectively, designed a CLS scheme with batch verification for V2I secure communication in VANETs using blockchain technology. Ali et al. [16] also proposed an efficient CLAS scheme based on bilinear pairing, which took advantage of the immutability and openness of blockchain and could verify whether the identities of all vehicles in VANETs was legitimate. Recently, Mei et al. [17] proposed an effective CLAS scheme with conditional privacy preservation, which used full aggregation technology to reduce bandwidth resources and computational overhead. However, since these schemes [15–17, 29, 30] are all implemented based on bilinear pairing, the process of signing and verifying has a lot of overhead.

With the arrival of the 5G era, the research on security and privacy preservation methods in 5G-enabled vehicular networks is gradually started [2, 3, 18, 19, 31, 32]. Eiza et al. [3] proposed a novel system model for 5G-enabled vehicular networks, which provided a reliable and secure real-time video reporting service, and also proposed a real-time video reporting service protocol that met security and privacy preservation requirements. Similarly, Zhang et al. [2] also proposed an authentication scheme based on edge computing, which used edge computing vehicles to realize communication and verification between vehicles without the participation of RSUs. Recently, Cui et al. [31] proposed

a reliable and efficient content sharing scheme for 5G-enabled vehicular networks. Wang et al. [32] proposed a hybrid D2D message authentication scheme for 5G-enabled vehicular networks.

3. Preliminaries

This section first introduces the theoretical basis of our proposed scheme. ECC and cuckoo filter are important components of the proposed CLAS scheme, in which ECC can ensure the efficient performance and security of the system, while the cuckoo filter has an efficient search feature to find malicious user quickly. After that, the system model and the authentication process are described. Then, the basic framework and execution process of the general CLAS scheme are given. Besides, the threat model elaborates the assumption conditions, attack model, and adversary model. Finally, the design goals are summarized.

3.1. ECC. ECC is widely used in the design of cryptographic protocols and security schemes because of its higher computational efficiency and communication efficiency.

Suppose that a finite field F_p is defined over a large prime p and an elliptic curve E defined over the field F_p is the set of solutions (x, y) to the equation $E : y^2 = x^3 + ax + b \bmod p$, where $a, b \in F_p$, $(x, y) \in F_p * F_p$, and $(4a^3 + 27b^2) \bmod p \neq 0$. Therefore, the elliptic curve E is the set of such solutions together with a point at infinity O, where O is identity element, and the points of E form a finite Abelian group. Here, p and E are fixed and publicly known. Furthermore, suppose that $P \in E$ is a fixed and publicly know point, and G_p is an additive group whose order is a large prime q and whose generator is P. Therefore, G_p is a finite cyclic subgroup on the elliptic curve E. A brief description of the two difficult problems of ECC is as follows.

(1) Elliptic curve discrete logarithm problem (ECDLP): for any given two random points $P, Q \in G_p$, where $x \in [0..q-1]$ is unknown and $Q = xP$ is known, it is hard to compute x in polynomial time with nonnegligible probability

(2) Computational Diffie–Hellman problem (CDHP): for any given two random points $aP, bP \in G_p$, where $a, b \in [0..q-1]$ are unknown, it is hard to computer $Q = abP$, $Q \in G_p$ in polynomial time with nonnegligible probability

The security of ECC is based on the difficulties of the ECDLP and CDHP.

3.2. Cuckoo Filter. A cuckoo filter is a compact variant of a cuckoo hash table and a probabilistic data structure for approximate set membership tests. It only stores fingerprints for each item inserted, supports dynamic addition and deletion of items, and provides higher lookup performance compared to standard Bloom filters without incurring higher overhead in space and performance. Therefore, cuckoo filter is suitable for applications that store many items and aim for low false-positive rates [33].

The cuckoo filter uses two hash functions $h_1(x)$ and $h_2(x)$ to insert a new item x into a hash table. If one of x's two locations is empty, then the algorithm just puts x in there. But if both are full, the algorithm randomly selects one of them, kicks out the existing item, and puts x in there. At the same time, the kicked item is put to its own alternate location. The insertion operation of the cuckoo filter is shown in Figure 1, where Figures 1(a) and 1(b) show the process of inserting a new item x into a hash table and Figure 1(c) shows an instance of a cuckoo filter.

3.3. System Model. As shown in Figure 2, the system model of 5G-enabled vehicular networks mainly consists of four entities: trusted authority (TA), application server (AS), RSUs (i.e., 5G-RSUs or 5G-BSs), and vehicles, and they communicate with each other via C-V2X. The system model can be divided into two layers, in which the upper network is composed of TA and AS and the lower network is composed of RSUs and vehicles equipped with OBUs.

TA: TA includes a KGC and a trace authority (TRA). TA uses secure wired channels to communicate with other facilities and is mainly responsible for initializing system parameters for each registered vehicle and RSU in 5G-enabled vehicular networks. TA can also conditionally trace and revoke malicious vehicles to meet the requirements of security and privacy preservation. KGC is responsible for generating a partial private key for each vehicle, while TRA is responsible for generating a pseudonym for each vehicle and can trace the vehicle's real identity through the pseudo identity of the vehicle. TA's executive agency is generally the government's traffic management centre (TMC)

AS: AS is an application server with strong computing and storage capabilities. AS uses secure wired channels to communicate with other facilities and is mainly responsible for providing large-scale computing and storage services for 5G-enabled vehicular networks to collect and analyse traffic-related data

RSUs: RSUs are located on both sides of the road and generally include 5G base stations and road side units. RSUs use V2I communication mode to receive and transmit vehicles' information. RSUs can also communicate with TA and AS using secure wired channels. Therefore, RSUs have the ability to store and forward data and cooperate with vehicles for data analysis. Moreover, RSUs also check the integrity and authenticity of the messages to fulfil security requirements

Vehicles: vehicles are the main carrier for collecting driving data and sensing information in vehicular networks. Vehicles can use V2V communication mode to exchange information with adjacent vehicles and can also use V2N communication mode to communicate with TA and AS

When vehicles join vehicular networks, TA initializes system parameters so that each vehicle can be assigned a pseudo identity and a corresponding public and private key pair. The vehicle must sign each message before sending it. After receiving the message signature pair, the RSU needs to verify the signature to ensure the authenticity of the

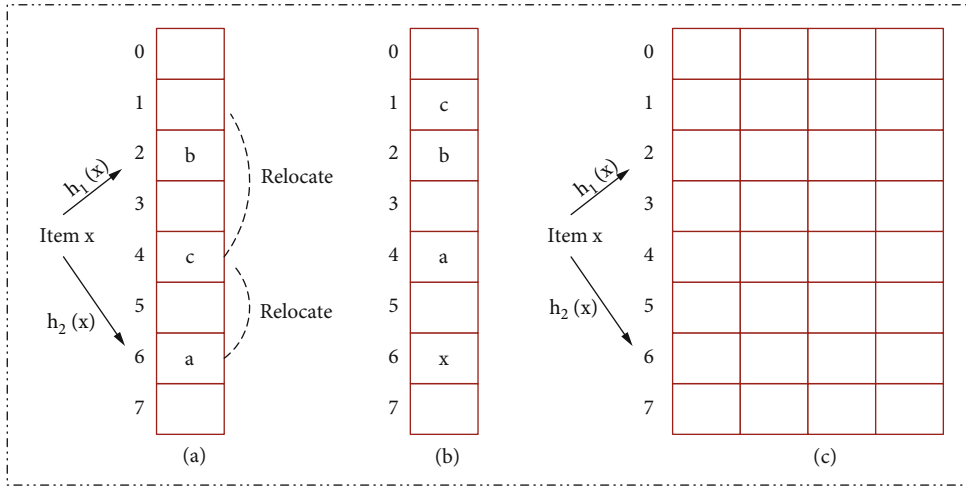

Figure 1: Insertion operation of cuckoo filter.

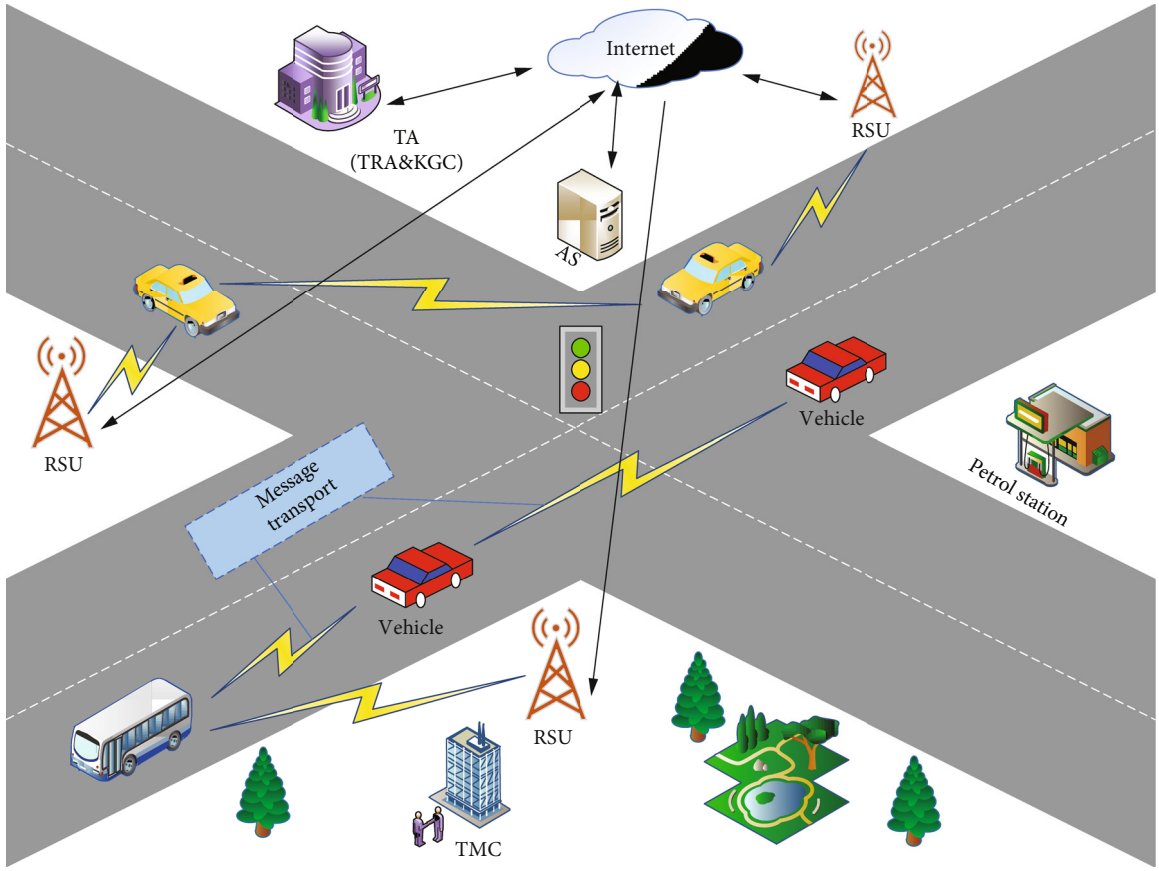

Figure 2: The system model of 5G-enabled vehicular networks.

signature and the integrity of the message. When the RSU receives a large number of traffic-related messages, the messages are aggregated, and the messages and signatures are sent to AS for verification and analysis. Finally, AS feeds back the results of authentication and analysis to the TA for further processing.

3.4. System Framework. Generally, the CLAS scheme consist of the following algorithms [11, 25]. According to the overview of our proposed scheme, the process of each algorithm is roughly described as follows:

(1) System initialization: according to the security parameter λ input by the system, TA generates an elliptic curve E, master secret keys x and s, system public keys K_{pub} and T_{pub}, and system parameters params, respectively. TA broadcasts system parameters params and keeps the master secret keys x and s

(2) Pseudo identity generation: vehicle V_i sends its real identity RID_i and its partial pseudo identity $PID_{i,1}$ to TRA. TRA verifies the authenticity of the vehicle's real identity RID_i and generates pseudonym PID_i and then sends it to KGC and vehicle V_i

(3) Partial private key generation: KGC takes system parameters params, master secret key x, and a vehicle's pseudo identity PID_i as input, returns vehicle's partial private key ppk_i, and sends it to the vehicle V_i over a secure channel

(4) Vehicle key generation: vehicle V_i takes its pseudo identity PID_i, partial private key ppk_i, state information ∇, and system parameters *params* as input and outputs a private key vsk_i, a public key vpk_i, a full public key PK_i, and a full private key sk_i

(5) Individual signature generation: this algorithm is performed by each vehicle V_i that takes state information ∇, its pseudo identity PID_i and a message M_i as input and responds with a signature σ_i as output

(6) Individual signature verification: this is an algorithm performed by a verifier, such as an RSU, that uses system parameters params, pseudo identity PID_i, message M_i, and signature σ_i to verify the validity of the signature σ_i

(7) Signature aggregation: this is an algorithm performed by an aggregate signature generator, such as an RSU, that aggregates n signatures σ_i of n messages M_i from n users V_i into a single short aggregate signature σ and send it to AS

(8) Aggregate signature verification: this algorithm is performed by AS for verifying the validity of the aggregate signature σ. It takes system parameters p *arams*, n pseudo identities $\{PID_1, PID_2, ...PID_n\}$, n messages $\{M_1, M_2, ...M_n\}$, and the aggregate signature σ as input and outputs true if the aggregate signature σ is valid and false otherwise

Based on the above algorithm process, the execution flow of our proposed scheme is shown in Figure 3.

3.5. Threat Model. The proposed CLAS scheme has the following assumptions:

(1) TA and AS are fully trusted, independent, and reliable entities with sufficient storage and computation capabilities

(2) Each RSU is an honest-but-curious entity with slightly less storage and computation capabilities than TA and AS and provides good network coverage and ultrafast information transmission speed

(3) Each vehicle is an untrusted entity, equipped with an OBU with limited storage and computation capabilities and equipped with a tamper proof device (TPD) to protect absolute data security

In order to prove that our proposed CLAS scheme is not existentially forgeable against adaptive chosen-message attack in ROM, we define two types of adversaries which are similar to these schemes [10, 12, 13, 34].

Type 1. $\mathcal{A}_1$ adversary is an external adversary who has the ability to launch a public key replacement attack but cannot obtain the master secret key or the partial private key

Type 2. $\mathcal{A}_2$ adversary represents a malicious-but-passive internal attacker who can access the master secret key and the partial private key but cannot replace any user's public key

We consider two games played by a challenger $\mathcal{C}$ and an adversary $\mathcal{A} \in \{\mathcal{A}_1, \mathcal{A}_2\}$ to define the security of our proposed scheme.

The adversary $\mathcal{A}$ can query the following oracles.

Create Vehicle (ID). When receiving a query from $\mathcal{A}$, $\mathcal{C}$ takes a vehicle's identity ID as input, calculates vpk_i, and sends it to $\mathcal{A}$

Partial Private Key (ID). When receiving the partial private key query for a vehicle whose identity is ID from $\mathcal{A}$, $\mathcal{C}$ sends ppk_i to $\mathcal{A}$

Public Key (ID). When $\mathcal{A}$ requests $\mathcal{C}$ for the public key of a vehicle whose identity is ID, $\mathcal{C}$ calculates PK_i and sends it to $\mathcal{A}$

Secret Key (ID). Given a vehicle whose identity is ID, the oracle sends vsk_i to $\mathcal{A}$

Replace Public Key(ID, PK_i^*). When receiving this query from $\mathcal{A}$, $\mathcal{C}$ replaces the public key PK_i with PK_i^*

Sign (ID, M_i). When receiving a message M_i from a vehicle whose identity is ID, $\mathcal{C}$ calculates a certificateless signature σ_i and sends it to $\mathcal{A}$

3.6. Security and Privacy Requirements. Considering that 5G-enabled vehicular networks must meet the requirements of security, efficiency, and privacy protection, the proposed CLAS scheme needs to satisfy the security and privacy requirements described as follows:

Message integrity and authentication: in 5G-enabled vehicular networks, when an RSU receives a signature and message sent by a vehicle, it must verify the authenticity and integrity of the message to ensure the legitimacy of the vehicle and to ensure that the message has not been tampered with, impersonated, or forged by a malicious attacker

Privacy preserving: during the authentication process for 5G-enabled vehicular networks, vehicles are not allowed to communicate using their real identities and must use pseudonyms

Traceability and revocability: when vehicles communicate with pseudonyms, they are likely to be attacked by malicious vehicles. Therefore, TRA must have the ability to obtain the real identity of the malicious vehicle in order to trace its malicious act, as well as put in place certain mechanisms for management, such as revoking the malicious vehicle

Unlinkability: an attacker must not be able to infer a vehicle from multiple messages sent by the same vehicle by cross-linking

Resistance to various attacks: 5G-enabled vehicular networks are vulnerable, so the proposed CLAS scheme must have the ability to resist various general attacks, such as

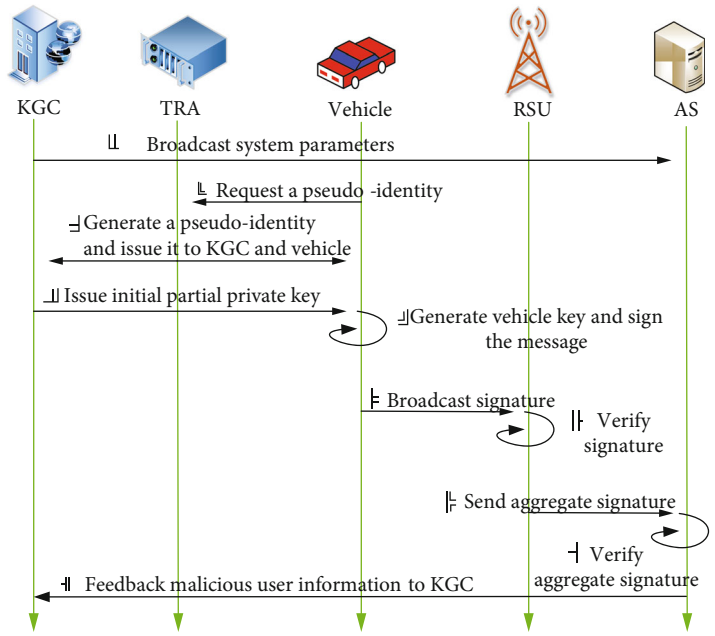

Figure 3: The execution flow of our proposed CLAS scheme for 5G-enabled vehicular networks.

impersonation attacks, replay attacks, modification attacks, and information injection attacks

4. The Proposed CLAS Scheme

In this section, we propose an efficient and secure CLAS scheme, which is implemented without bilinear pairing and map to point hash operations. Moreover, when the aggregate signature is verified to be invalid, the proposed CLAS scheme uses binary search to identify invalid signatures and introduces a cuckoo filter to revoke malicious vehicles to prevent the attack again. In addition, the proposed CLAS scheme also uses a random vector to resist information injection attacks. This scheme includes eleven phases: system initialization, pseudo identity generation, vehicle registration, partial private key generation, vehicle key generation, individual signature generation, individual signature verification, signature aggregation, aggregate signature verification, invalid signature identification, and malicious vehicle revocation. The main symbols and description used in this scheme are shown in Table 1.

The detailed implement process of each phase of the proposed CLAS scheme is described as follows.

(i) System initialization: in this phase, KGC and TRA initialize system parameters for RSUs and vehicles

(1) According to the security parameter λ, KGC selects two large prime numbers p, q, respectively, and generates an elliptic curve $E: y^2 = x^3 + ax + b \bmod p$, where $p > 3$, $a, b \in F_p$, $(x, y) \in F_p * F_p$, and $(4a^3 + 27b^2) \bmod p \neq 0$

(2) KGC selects a point P on the elliptic curve E as its random generator and generates a group G with

Table 1: Symbols and description.

Symbols	Description
p, q	Two large primes
E	An elliptic curve : $y^2 = x^3 + ax + b \bmod p$
G	A cyclic additive group
P	A generator of the group G
TA	A trusted authority
KGC	A key generation centre
TRA	A trace authority
V_i	The i_{th} vehicle
RSU	A road side unit
OBU	An onboard unit
K_{pub}, x	The public/private key pair of the KGC
T_{pub}, s	The public/private key pair of the TRA
h_1, h_2, h_3, h_4	Four one-way collision-resistant hash functions
ppk_i	The partial private key of the vehicle V_i
$\text{vsk}_i, \text{vpk}_i$	The public/private key pair of the vehicle V_i
PID_i	The pseudo identity of the vehicle V_i
RID_i	The real identity of the vehicle V_i
VP_i	A valid period of the pseudo identity
t_i	The latest time stamp
$\oplus$	The bit-wise exclusive-OR (XOR) operation
$\|\|$	The message concatenation operation
M_i	A traffic-related sensing message
σ_i	A certificateless signature on a message M_i

order q from P. TRA also selects a point P on the elliptic curve E as its random generator and generates a group G with order q from P

(3) KGC selects a random number $x \in Z_q^*$ from the finite field as its master private key, which is used to extract the partial private key ppk_i. Then, it calculates the corresponding public key $K_{\text{pub}} = x \cdot P$, where x is only known by KGC

(4) TRA selects a random number $s \in Z_q^*$ as its master private key for traceability and calculates the system public key $T_{\text{pub}} = s \cdot P$, where s is only known by TRA

(5) KGC and TRA choose four secure hash functions: h_1, h_2, h_3, and h_4, where $h_1 : \{0,1\}^* \longrightarrow Z_q^*$, $h_2 : \{0,1\}^* \times G \longrightarrow Z_q^*$, $h_3 : \{0,1\}^* \times G \times G \longrightarrow Z_q^*$, and $h_4 : \{0,1\}^* \times G \times G \times G \longrightarrow Z_q^*$

(6) KGC and TRA keep the master secrets key x and s and issue the system parameters:

$$\text{params} = \{P, p, q, E, G, h_1, h_2, h_3, h_4, K_{\text{pub}}, T_{\text{pub}}\} \quad (1)$$

Any vehicle that has successfully registered with the TA can access the system parameters via a secure channel and store them in its TPD. Similarly, any RSU can also access the system parameters after successful registration.

(ii) Pseudo identity generation: in this phase, TRA works with vehicles to generate pseudo identities for vehicles to conditionally preserve their privacy, which allows them to send messages anonymously

(1) The vehicle V_i with its real identity RID_i picks a random number $\xi_i \in Z_q^*$ to calculate the its partial pseudo identity $\text{PID}_{i,1} = \xi_i \cdot P$, and

$$A_i = \text{RID}_i \oplus h_1\left(\xi_i \cdot \text{PID}_{i,1} \| T_{\text{pub}}\right) \cdot \xi_i \cdot T_{\text{pub}} \quad (2)$$

The vehicle V_i then sends $\{\text{PID}_{i,1}, A_i\}$ securely to TRA.

(2) When TRA receives the tuple $\{PID_{i,1}, A_i\}$ from the vehicle V_i, it first checks whether

$$\text{RID}_i = A_i \oplus h_1\left(\xi_i \cdot \text{PID}_{i,1} \| T_{\text{pub}}\right) \cdot \xi_i \cdot T_{\text{pub}} \quad (3)$$

is valid or not. If the identity of the vehicle V_i fails, the TRA will discard the tuple; otherwise, it will calculate

$$\text{PID}_{i,2} = \text{RID}_i \oplus h_1(s \cdot \text{PID}_{i,1} \| \text{VP}_i \| \text{params}) \quad (4)$$

and generate the vehicle's pseudo identity $\text{PID}_i = \{\text{PID}_{i,1}, \text{PID}_{i,2}, \text{VP}_i\}$, where VP_i is the valid period of the vehicle's pseudo identity. Finally, TRA sends PID_i to KGC and the vehicle V_i through a secure channel

(iii) Vehicle registration: after receiving the pseudo identity PID_i, the vehicle V_i stores it in its TPD and then communicates with other entities using the pseudo identity PID_i in 5G-enabled vehicular networks. To preserve vehicle's privacy and trace malicious vehicle, TRA stores both the real identity RID_i and the pseudo identity PID_i of the vehicle V_i. Once a vehicle or RSU reports a malicious vehicle, TRA can obtain the real identity of the vehicle from the vehicle's pseudo identity according to the master private key, and TRA inserts the malicious vehicle's pseudo identity PID_i into the negative cuckoo filter (RPID-cuckoo) to revoke it

(iv) Partial private key generation: in this phase, KGC generates a partial private key based on the received pseudo identity of the vehicle. The preload method is also used to store the pseudo identities of the vehicle and the partial private keys of the vehicle and to reload when them needs to be updated

(1) After receiving the pseudo identity PID_i of the vehicle V_i, KGC checks the validity of the VP_i in PID_i and queries the RPID-cuckoo filter to ensure that the pseudo identity of the vehicle has not been revoked by the TRA

(2) If the pseudo identity has not expired and the vehicle has not been revoked, KGC chooses a random number $u_i \in Z_q^*$ to calculate $U_i = u_i \cdot P$, $h_{2i} = h_2(\text{PID}_i \| K_{\text{pub}} \| \text{params})$ and creates a partial private key $\text{ppk}_i = u_i + \text{xh}_{2i} (\text{mod } q)$ for the vehicle V_i

(3) KGC sends the tuple (U_i, ppk_i) to the vehicle V_i through a secure channel

(v) Vehicle key generation: in this phase, the vehicle generates a partial public/private key pair and creates its full public/private key pair

(1) After receiving the tuple (U_i, ppk_i) from KGC, the vehicle V_i calculates $h_{2i} = h_2(\text{PID}_i \| K_{\text{pub}} \| \text{params})$ and checks the validity of the partial private key ppk_i by calculating whether the equation

$$\text{ppk}_i \cdot P = u_i \cdot P + xh_{2i} \cdot P \,(\text{mod } q) = U_i + h_{2i} K_{\text{pub}} \,(\text{mod } q) \quad (5)$$

holds or not. If it holds, the vehicle V_i stores the partial private key ppk_i in its TPD; otherwise, the vehicle V_i discards the partial private key ppk_i

(2) The vehicle V_i chooses two random numbers $a_i, b_i \in Z_q^*$ and calculates $h_{3i} = h_3(\text{PID}_i \| \nabla \| \text{params})$, $\theta_i = a_i h_{3i}$, $B_i = b_i \cdot P$, and $C_i = \theta_i \cdot P$, respectively

(3) The vehicle V_i selects θ_i as the vehicle private key vsk_i, $(C_i + U_i)$ as the vehicle public key vpk_i and forms the vehicle's full private key $\text{sk}_i = (\text{ppk}_i, \text{vsk}_i)$ and full public key $\text{PK}_i = (U_i, \text{vpk}_i)$

(vi) Individual signature generation: in this phase, the vehicle signs each traffic-related message that is about to be sent to meet authentication and message integrity requirements

(1) The vehicle V_i randomly selects a pseudo identity PID_i from its TPD and picks a latest timestamp t_i to prevent replay attacks and ensure the timeliness of sensing information collection

(2) The vehicle V_i picks a random number $r_i \in Z_q^*$, computes $R_i = r_i \cdot B_i$, $h_{4i} = h_4(M_i \| \mathrm{PID}_i \| \mathrm{vpk}_i \| R_i \| \nabla \| t_i)$,

$$S_i = r_i b_i + h_{4i}(\theta_i + \mathrm{ppk}_i) \ (\mathrm{mod}\ q), \tag{6}$$

and then constructs a certificateless signature $\sigma_i = (R_i, S_i)$ on the traffic-related sensing message M_i

(3) The signature-message tuple $msg = \{M_i \| \mathrm{PID}_i \| \mathrm{vpk}_i \| \sigma_i \| t_i\}$ is broadcasted to the nearby RSUs or other vehicles for verification

(vii) Individual signature verification: in this phase, each RSU or vehicle that receive the signature-message tuple is responsible for verifying the validity of the individual signature

(1) After receiving the signature-message tuple msg, each RSU or vehicle acts as a verifier to check whether that both t_i in msg and VP_i in PID_i are valid, and as long as one of them is invalid, the verifier can reject the message and stop verification

(2) The verifier looks up the RPID-cuckoo filter to ensure that the PID_i has not been revoked by TRA. If the vehicle V_i has not been revoked, the verifier performs the following procedures

(3) The verifier computes $h_{2i} = h_2(\mathrm{PID}_i \| K_{\mathrm{pub}} \| \mathrm{params})$ and $h_{4i} = h_4(M_i \| \mathrm{PID}_i \| vpk_i \| R_i \| \nabla \| t_i)$, respectively, and checks the validity of the signature σ_i by verifying whether the equation

$$S_i \cdot P = R_i + h_{4i}\left(\mathrm{vpk}_i + h_{2i} K_{\mathrm{pub}}\right), \tag{7}$$

holds or not. If it holds, the verifier accepts the signature-message tuple msg

(viii) Signature aggregation: in this phase, each RSU acts as an aggregate signature generator. When receiving multiple signature-message tuples $\{M_1 \| \mathrm{PID}_1 \| \mathrm{vpk}_1 \| \sigma_1 \| t_1\}$, $\{M_2 \| \mathrm{PID}_2 \| \mathrm{vpk}_2 \| \sigma_2 \| t_2\}$, …, $\{M_n \| \mathrm{PID}_n \| \mathrm{vpk}_n \| \sigma_n \| t_n\}$ from n vehicles $\{V_1, V_2, \ldots. V_n\}$, the aggregate signature generator aggregates multiple certificateless signatures into a single short signature, which can both reduce the communication overhead and make full use of the computational resources of the RSU. In this phase, the adversary can launch an injection attack by tampering with several valid signatures [29]. In order to prevent information injection attacks in the aggregation signature phase, we use the random vector to resist this attack

(1) The aggregate signature generator randomly generates a vector $v = \{v_1, v_2, v_3 \cdots, v_n\}$ that is used to resist information injection attacks, where each v_i is a random exponent in the range $[1, 2^t]$ and t is a very small integer

(2) The aggregate signature generator calculates $S = \sum_{i=1}^{n} v_i S_i$, $R = \sum_{i=1}^{n} v_i R_i$, and $C = \sum_{i=1}^{n} h_{4i} v_i vpk_i$, respectively

(3) The aggregate signature generator outputs $\sigma = (R, S)$ as the certificateless aggregate signature and sends it to the AS

(ix) Aggregate signature verification: in this phase, AS is responsible for verifying the validity of the certificateless aggregate signature aggregated by the aggregate signature generator

(1) AS checks whether t_i in msg and VP_i in PID_i are both valid for $i = 1, 2, 3, \cdots, n$. If they are all valid, then AS performs the following procedures

(2) AS looks up the RPID-cuckoo filter to ensure n pseudo identities $\{\mathrm{PID}_1, \mathrm{PID}_2, \ldots.\mathrm{PID}_n\}$ have not been revoked by TRA. If a set of vehicles $\{V_1, V_2, \cdots V_n\}$ has not been revoked, AS performs the following procedures

(3) AS computes $h_{2i} = h_2(\mathrm{PID}_i \| K_{\mathrm{pub}} \| \mathrm{params})$ and $h_{4i} = h_4(M_i \| \mathrm{PID}_i \| vpk_i \| R_i \| \nabla \| t_i)$ for $i = 1, 2, 3, \cdots, n$

(4) AS checks whether the following equation holds or not

$$S \cdot P = R + C + \sum_{i=1}^{n} v_i h_{4i} h_{2i} K_{\mathrm{pub}}. \tag{8}$$

If it holds, then AS accepts the certificateless aggregate signature; otherwise, AS performs the following phases

All of the above phases are shown in Figure 4.

(x) Invalid signature identification: in this phase, when an attacker inserts an invalid signature into the aggregate signature in order to interfere with the verification resulting in invalid aggregate signature verification, AS identifies invalid signatures from the aggregate signature, requests TRA to revoke the malicious vehicles and broadcast them, and accepts the remaining valid signatures in order to reduce the computational overhead caused by repeated verification and prevent malicious vehicles from attacking. In the process of invalid signature identification, binary search is used jointly by AS and RSU

(1) The RSU sorts the previously received multiple signatures, finds the middle position, divides the multiple signatures into two sets and then performs

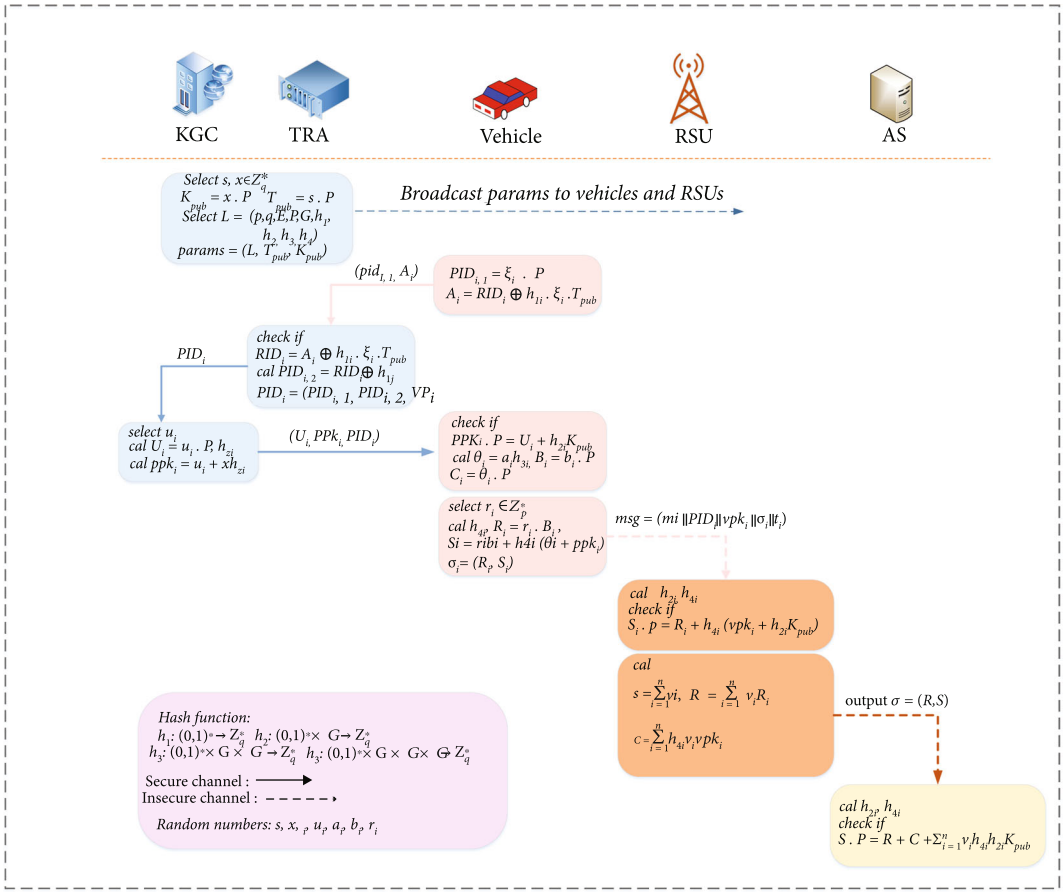

Figure 4: The detailed algorithm of our proposed CLAS scheme.

aggregate signature on them, respectively, and sends the two aggregate signatures to AS for verification

(2) If AS verifies that either of the two aggregate signatures is invalid, the above procedures 1 and 2 are repeated for the invalid aggregate signatures until an invalid signature is found

(3) AS sends multiple pseudo identities corresponding to invalid signatures $\{\mathrm{PID}_i, t_i\}$ to TRA to trace the real identity RID_i of malicious vehicles and revoke them while sending also multiple pseudo identities corresponding to valid signatures $\{\mathrm{PID}_i, \mathrm{t}_i\}$ to TRA and receiving these valid signatures

The overall invalid signature identification steps are shown in Algorithm 1

(xi) Malicious vehicle revocation: in this phase, after AS identifies invalid and valid signatures from the aggregate signature and sends them to TRA, TRA traces the real identity RID_i of the malicious vehicle through the pseudo identity PID_i corresponding to the invalid signature and then maps out all preloaded pseudo identities of the vehicle V_i. Finally, TRA makes use of the cuckoo filter to revoke the malicious vehicles. Moreover, when the fingerprint of each vehicle is known, in order to improve the efficiency of signature verification, a cuckoo filter is used to assist in verifying the signature. At the same time, in order to reduce false positives, two cuckoo filters are used to store relative fingerprint information: the positive filter (PID-cuckoo) is used to store fingerprints with valid signatures, and the negative filter (RPID-cuckoo) is used to store fingerprints with invalid signatures

(1) TRA generates a fingerprint $f_i = \mathrm{Fingerprint}(L_{\mathrm{inv}})$ and stores it in the RPID-cuckoo, where $L_{\mathrm{inv}} = \{\mathrm{P}$ $\mathrm{ID}_i\}$ is a list of all the preloaded pseudo identities of the malicious vehicle V_i. Then, TRA generates a fingerprint $f_i = \mathrm{Fingerprint}(L_{\mathrm{valid}})$ and stores it in the PID-cuckoo, where $L_{\mathrm{valid}} = \{\mathrm{PID}_i\}$ is a list that stores the current pseudo identity of the valid vehicle V_i

(2) TRA generates a notification message and sends the received aggregation signature verification result through RSU to all vehicles in its range. In order to save communication resources, TRA only broadcasts the negative cuckoo filter results, except for vehicle and RSU active queries. The specific implementation of the notification message is shown in Algorithm 2

(3) When the receiver of the message M_i proactively verifies the pseudo identity PID_i of the vehicle V_i

```
//List represents the aggregate signature that needs to be verified
//invList represents invalid signatures that have been identified
1:  if aggregate-verify(List,low,high) == true then
2:     return
3:  else
4:     if low == high then
5:     invList.ppend(ist[low])
6:     return
7:     else
8:     mid=(high+low)/2
9:     InvalidSignature-dentification(List,invList,low,mid)
10:     InvalidSignature-dentification(List,invList,mid+1,high)
11:     //return 1
12:     end if
13:  end if
```

Algorithm 1:Invalid signature identification (List, invList, low, and high).

```
//validList represents valid signatures that have been identified
//invList represents invalid signatures that have been identified
1:  for PID_i ∈ validList(V_i) do
2:     L_valid ⟵ {PID_i}
3:     PID – Cuckoo.insert(L_valid)
4:  end for
5:  for PID_i ∈ invList(V_i) do
6:     L_inv ⟵ {PID_i}
7:     RPID – Cuckoo.insert(L_inv)
8:  end for
9:  return {RPID – Cuckoo, Sig_s(RPID – Cuckoo)}
```

Algorithm 2: Notification message generation.

Table 2: The query results of the cuckoo filter.

Case	Positive filter	Negative filter	Result
1	True	False	Valid
2	False	True	Invalid
3	False	False	Wait for next broadcast
4	True	True	Wait for reconfirmation

to determine whether the vehicle V_i is a malicious vehicle, it calculates the fingerprint f_i = Fingerprint (L_i) corresponding to the vehicle V_i, where $L_i = \{\text{P ID}_i\}$, then lookups the RPID-cuckoo and the PID-cuckoo, respectively, and obtains the query result. The four possible query results are listed in Table 2

Both case 1 and case 2 explicitly state whether the pseudo identity PID_i is revoked or not. There is a certain probability that the result of the query is case 3. Case 3 indicates that AS has not yet verified the pseudo identity PID_i, or the verification result has not been updated to the cuckoo filter in time. Therefore, the signature verification of the vehicle needs to wait for the next round of cuckoo filter update. However, if the number of queries exceeds the preset number of rounds of the system and the query still fails, the message receiver will check whether the message M_i is valid or not according to equation (7). Case 4 occurs because the cuckoo filter has a certain false-positive rate. Therefore, the message receiver needs to send a reconfirmation message to TRA via RSU. The process of the message receiver querying the cuckoo filter to verify the pseudo identity PID_i is shown in Algorithm 3

5. Security Proof and Analysis

In this section, we explain the correctness of the verification process in our proposed scheme and prove the unforgeability of the signature in the proposed CLAS scheme. Finally, we analyse how the proposed CLAS scheme meets the requirements of security and privacy protection.

5.1. Correctness Proof. We need to explain why signature is verified to be valid if and only if equations (7) and (8) hold. The proof derivation goes as follows.

Require: $L_i \longleftarrow \{PID_i\}$
1: Receiver calculates a $Fingerprintf_i = Fingerprint(L_i)$
2: **While** VP is valid
3: Receiver queries *RP*ID – Cuckoo and *P*ID – Cuckoo with query
4: **if** $f_i \in PID - Cuckoo$ **then**
5: **if** $f_i \notin RPID - Cuckoo$ **then**
6: Receiver continues to execute the signature verification; **break;**
7: **else**
8: **if** $f_i \in RPID - Cuckoo$ **then**
9: Receiver re-confirmation needed; **break;**
10: **end if**
11: **end if**
12: **else**
13: **if** $f_i \in RPID - Cuckoo$ **then**
14: Receiver stops executing signature verification; **break;**
15: **end if**
16: **if** $f_i \notin RPID - Cuckoo$ **then**
17: Receiver wait for next broadcast; **break;**
18: **end if**
19: **end if**
20: **end while**

Algorithm 3: Cuckoo filter query.

(1) Individual signature verification:

$$\begin{aligned} L.H.S &= S_i \cdot \mathrm{P} \\ &= (r_i b_i + h_{4i}(\theta_i + \mathrm{ppk}_i)) \cdot \mathrm{P} \\ &= r_i b_i \cdot \mathrm{P} + h_{4i}(\theta_i \cdot \mathrm{P} + \mathrm{ppk}_i \cdot \mathrm{P}) \\ &= \mathrm{r}_i B_i + h_{4i}(C_i + (u_i + \mathrm{xh}_{2i}) \cdot \mathrm{P}) \\ &= \mathrm{r}_i B_i + h_{4i}\left(C_i + U_i + h_{2i} K_{\mathrm{pub}}\right) \\ &= R_i + h_{4i}\left(\mathrm{vpk}_i + h_{2i} K_{\mathrm{pub}}\right) \\ &= R.H.S. \end{aligned} \tag{9}$$

(2) Aggregate signature verification:

$$\begin{aligned} L.H.S &= S \cdot \mathrm{P} \\ &= \left(\sum_{i=1}^{n} v_i S_i\right) \cdot \mathrm{P} \\ &= \left(\sum_{i=1}^{n} v_i (r_i b_i + h_{4i}(\theta_i + \mathrm{ppk}_i))\right) \cdot \mathrm{P} \\ &= \left(\sum_{i=1}^{n} v_i \mathrm{r}_i B_i\right) + \left(\sum_{i=1}^{n} v_i h_{4i} C_i\right) + \left(\sum_{i=1}^{n} v_i h_{4i}(u_i + \mathrm{xh}_{2i})\right) \cdot \mathrm{P} \\ &= \left(\sum_{i=1}^{n} v_i \mathrm{r}_i B_i\right) + \left(\sum_{i=1}^{n} v_i h_{4i} C_i\right) + \left(\sum_{i=1}^{n} v_i U_i\right) \\ &\quad + \left(\sum_{i=1}^{n} v_i h_{4i} h_{2i}\right) \cdot K_{\mathrm{pub}} \\ &= \sum_{i=1}^{n} v_i R_i + \sum_{i=1}^{n} v_i h_{4i} \mathrm{vpk}_i + \sum_{i=1}^{n} v_i h_{4i} h_{2i} K_{\mathrm{pub}} \\ &= R + C + \sum_{i=1}^{n} v_i h_{4i} h_{2i} K_{\mathrm{pub}} = R.H.S. \end{aligned} \tag{10}$$

5.2. Security Proof. In this section, we provide formal security proof for the proposed CLAS scheme for 5G-enabled vehicular networks. As mentioned earlier, in order to prove that our proposed scheme is existentially unforgeable against an adaptive chosen message attack under the ROM, we define two types of adversaries which are similar to [10, 12, 13, 34] and consider two games.

Theorem 1. *In the ROM, if there is a polynomial time Type 1 adversary $\mathscr{A}_1$ who can forge a valid signature of our scheme in an attack model of game 1 after making at most q_h times queries to the random oracles h_iQuery, q_c times Create Vehicle (ID) queries, and q_s times Sign (ID, M_i) queries, that is, win the game with an nonnegligible probability* Pr [*Succ* $(\mathscr{A}_1)$]*, then there must be a polynomial time challenger $\mathscr{C}$ that can solve the ECDLP with a nonnegligible advantage ε.*

Proof. In the ROM, it is assumed that there is a probabilistic polynomial time adversary $\mathscr{A}_1$ who has enough ability to forge the signature-message tuple $\mathrm{msg} = \{M_i \| \mathrm{PID}_i \| \mathrm{vpk}_i \| \sigma_i \| t_i\}$ of the user ID_τ. Given a random instance $(P, Q = s \cdot P)$ of the ECDLP to compute s, a polynomial time challenger $\mathscr{C}$ calls $\mathscr{A}_1$ as its subroutine and solves the ECDLP with a nonnegligible advantage in polynomial time.

$\mathscr{A}_1$ performs the following queries:

Setup (ID). $\mathscr{C}$ inputs the security parameter λ in the system initialization phase and then sets $K_{\mathrm{pub}} = Q$ and sends the system parameter $\mathrm{params} = \{P, p, q, E, G, h_2, h_3, h_4, K_{\mathrm{pub}}, T_{\mathrm{pub}}\}$ to $\mathscr{A}_1$. In this process, $\mathscr{C}$ constructs and maintains five hash lists $L_{h_{2i}}$, $L_{h_{3i}}$, $L_{h_{4i}}$, L_u, and L_{pk}, all of which are initialized to empty

Create Vehicle (ID). $\mathscr{C}$ maintains a list $L_u = \{(\mathrm{ID}, U_i, \mathrm{vpk}_i, \mathrm{ppk}_i, \mathrm{vsk}_i, \mathrm{h}_{2i})\}$. When $\mathscr{A}_1$ makes a query on ID, and if ID is in L_u, then $\mathscr{C}$ sends $(\mathrm{ID}, U_i, \mathrm{vpk}_i, \mathrm{ppk}_i, \mathrm{vsk}_i, h_2)$ to

$\mathscr{A}_1$. Otherwise, $\mathscr{C}$ selects three random numbers $x, y, z \in Z_q^*$ and calculates $U_i = xK_{\text{pub}} + \text{x}yP$, $vpk_i = \text{x}zP + xy + xK_{\text{pub}}$, $ppk_i = xz$, $vsk_i = xy$, and $h_{2i} = h_2(ID \| K_{\text{pub}} \| \text{params}) \longleftarrow -x \pmod q$. Finally, $\mathscr{C}$ sends $(\text{ID}, U_i, \text{vpk}_i, \text{ppk}_i, \text{vsk}_i, h_2)$ to $\mathscr{A}_1$ and inserts $(\text{ID}, K_{\text{pub}}, \text{params}, h_{2i})$ into L_{h_2}

h_{2i} Query. $\mathscr{C}$ maintains a list $L_{h_2} = \{(\text{ID}, K_{\text{pub}}, \text{params}, h_{2i})\}$. When $\mathscr{C}$ receives a query $(\text{ID}, K_{\text{pub}}, \text{params})$ from $\mathscr{A}_1$, if ID is in L_{h_2}, then it sends h_{2i} to $\mathscr{A}_1$. Otherwise, $\mathscr{C}$ executes Create Vehicle (ID) to calculate $h_{2i} = h_2(\text{ID} \| K_{\text{pub}} \| \text{params})$ and then sends it to $\mathscr{A}_1$

h_{3i} Query. $\mathscr{C}$ maintains a list $L_{h_{3i}} = \{(\text{ID}, \text{params}, \nabla, h_{3i})\}$. When $\mathscr{C}$ receives a query $(\text{ID}, \text{params}, \nabla)$ from A_1, if $L_{h_{3i}}$ contains $(\text{ID}, \text{params}, \nabla, h_{3i})$, then $\mathscr{C}$ sends h_{3i} to $\mathscr{A}_1$. Otherwise, $\mathscr{C}$ chooses a random number $h_{3i} \in Z_q^*$, calculates $h_{3i} = h_3(\text{ID} \| \nabla \| params)$, sends h_{3i} to $\mathscr{A}_1$, and inserts $(\text{ID}, params, \nabla, h_{3i})$ into $L_{h_{3i}}$

h_{4i} Query. $\mathscr{C}$ maintains a list $L_{h_{4i}} = (\text{ID}, M_i, \text{vpk}_i, R_i, \nabla, t_i, h_{4i})$. When $\mathscr{A}_1$ makes a query on ID, and if ID is in $L_{h_{4i}}$, then $\mathscr{C}$ sends h_{4i} to $\mathscr{A}_1$. Otherwise, $\mathscr{C}$ selects a random number $h_{4i} \in Z_q^*$, calculates $h_{4i} = h_4(M_i \| \text{ID} \| \text{vpk}_i \| R_i \| \nabla \| t_i)$, sends h_{4i} to $\mathscr{A}_1$, and inserts $(ID, M_i, \text{vpk}_i, R_i, \nabla, t_i, h_{4i})$ into $L_{h_{4i}}$

Partial Private Key (ID). When $\mathscr{C}$ receives a partial private key query from $\mathscr{A}_1$ for ID, if $\text{ID} \neq \text{ID}_\tau$, $\mathscr{C}$ checks whether ID is already in L_u or not. If ID is in L_u, $\mathscr{C}$ sends ppk_i to $\mathscr{A}_1$. Otherwise, $\mathscr{C}$ runs Create Vehicle (ID) to obtain ppk_i and sends it to $\mathscr{A}_1$. In addition, if $\text{ID} = \text{ID}_\tau$, $\mathscr{C}$ stops the game

Public Key (ID). When $\mathscr{C}$ receives a public key query from $\mathscr{A}_1$ for ID, $\mathscr{C}$ checks whether ID exists in L_u or not. If it exists, $\mathscr{C}$ sends $\text{PK}_i = \{(\text{U}_i, \text{vpk}_i)\}$ to $\mathscr{A}_1$. Otherwise, $\mathscr{C}$ executes Create Vehicle (ID) to obtain the tuple (U_i, vpk_i) and sends it to $\mathscr{A}_1$

Replace Public Key $(\text{ID}, \text{PK}_i^*)$. $\mathscr{C}$ maintains a list $L_{\text{pk}} = (\text{ID}, u_i, U_i, \text{vsk}_i, \text{vpk}_i)$. When $\mathscr{A}_1$ makes a public key replacement query with $(\text{ID}, \text{PK}_i^*)$, where $U_i^* = u_i^* \cdot P$, $\text{vp k}_i^* = \text{vsk}_i^* \cdot P$ and $\text{PK}_i^* = (U_i^*, \text{vpk}_i^*)$, $\mathscr{C}$ sets $U_i = U_i^*$, $\text{vp k}_i = \text{vpk}_i^*$, $\text{vsk}_i = \text{vsk}_i^*$, and $\text{ppk}_i = \perp$, and inserts $(\text{ID}, u_i^*, U_i^*, \text{vsk}_i^*, \text{vpk}_i^*)$ to L_{pk}

Sign (ID, M_i). When $\mathscr{A}_1$ makes a signature query with (ID, M_i), $\mathscr{C}$ performs the following steps:

(1) If ID is in L_{pk}, $\mathscr{C}$ picks three random numbers x, y, $z \in Z_q^*$ such that $S_i = x$, $h_{4i} = h_4(M_i \| ID \| \text{vpk}_i \| R_i \| \nabla \| t_i) \longleftarrow x^{-1} \bmod q$ and $R_i = (x - y - z) \cdot P$, then inserts $(\text{ID}, M_i, \text{vpk}_i, R_i, \nabla, t_i, h_{4i})$ to $L_{h_{4i}}$, and finally sends the signature $\sigma_i = (R_i, S_i)$ to $\mathscr{A}_1$

(2) If ID is not in L_{pk}, because $\mathscr{C}$ knows the private key of the vehicle with ID, $\mathscr{C}$ acts like the procedure of the scheme

Forgery. In the end, $\mathscr{A}_1$ outputs a forged but valid certificateless signature $\sigma_i = (R_i, S_i)$ on (ID, M_i), which passes the signature verification phase. If $\text{ID} \neq \text{ID}_\tau$, $\mathscr{C}$ fails and stops

According to the forking lemma [35], $\mathscr{A}_1$ can obtain another forged valid signature $\sigma_i^* = (R_i^*, S_i^*)$ by simulating the game again with the same random tape but different choice of h_{4i} in polynomial time. We can get s according to the following equation; that is, $\mathscr{C}$ successfully solves the ECDLP. Then, we have

$$\begin{aligned} S_i \cdot P &= R_i + h_{4i}\left(\text{vpk}_i + h_{2i}K_{\text{pub}}\right), \\ S_i^* \cdot P &= R_i + h_{4i}^*\left(\text{vpk}_i + h_{2i}K_{\text{pub}}\right). \end{aligned} \tag{11}$$

From these two linear equations, we can derive the value s by

$$\begin{aligned} (S_i - S_i^*) \cdot P &= \left(R_i + h_{4i}\left(vpk_i + h_{2i}K_{pub}\right)\right) \\ &\quad - \left(R_i + h_{4i}^*\left(vpk_i + h_{2i}K_{pub}\right)\right) \Rightarrow (S_i - S_i^*) \cdot P \\ &= (h_{4i} - h_{4i}^*) \cdot h_{2i} \cdot s \cdot P \Rightarrow (S_i - S_i^*) \\ &= (h_{4i} - h_{4i}^*) \cdot h_{2i} \cdot s \Rightarrow s = \frac{(S_i - S_i^*)}{(h_{4i} - h_{4i}^*) \cdot h_{2i}}. \end{aligned} \tag{12}$$

Probabilistic Analysis. We analyse the advantages of $\mathscr{C}$ successfully obtaining s from $(P, Q = s \cdot P)$ to solve the ECDLP

In the process of executing Create Vehicle (ID) query, the random oracle assignment $h_2(\text{ID} \| K_{\text{pub}} \| \text{params})$ causes inconsistency, which happens with probability at most q_h/q. Therefore, the probability of successful simulation of q_c times is at least $(1 - q_h/q)^{q_c} \geq 1 - q_h q_c/q$. And, the probability of successful simulation of q_h times is at least $(1 - q_h/q)^{q_h} \geq 1 - {q_h}^2/q$. In addition, $\text{ID} = \text{ID}_\tau$ has a probability of $1/q_c$. Therefore, the overall probability of successful simulation is $\Pr[\text{Succ}(\mathscr{A}_1)] \geq (1 - q_h q_c/q)(1 - {q_h}^2/q)(1/q_c)\varepsilon$. It should be noted here that the time complexity $t + O(q_c + \text{q}_s)S$ of $\mathscr{C}$ is determined by the exponentiations executed in the Create Vehicle (ID) and Sign (ID, M_i) queries, where S is the time of a scalar multiplication operation.

$\mathscr{C}$ can successfully obtain s from $(P, Q = s \cdot P)$ with an advantage $(1 - q_h q_c/q)(1 - {q_h}^2/q)(1/q_c)\varepsilon$, where the time complexity of algorithm is $t + O(q_c + \text{q}_s)S$, which contradicts the ECDLP assumption. Therefore, the proposed scheme can resist the forgery attack of type 1 adversary $\mathscr{A}_1$ under the ROM. □

Theorem 2. *In the ROM, if there is a polynomial time Type 2 adversary $\mathscr{A}_2$ who can forge a valid signature of our scheme in an attack model of Game 2 after making at most q_h times queries to the random oracles h_i Query, q_c times Create Vehicle (ID) queries, and q_s times Sign (ID, M_i) queries, that is, win the game with an non-negligible probability* $\Pr[Succ(\mathscr{A}_2)]$*, then there must be a polynomial time challenger $\mathscr{C}$ that can solve the ECDLP with a nonnegligible advantage ε.*

Proof. In the ROM, it is assumed that there is a probabilistic polynomial time adversary $\mathscr{A}_2$ who has enough ability to forge the signature-message tuple $\text{msg} = \{M_i \| \text{PID}_i \| \text{vpk}_i \| \sigma_i \| t_i\}$ of the user ID_τ. Given a random instance $(P, Q = t \cdot P)$ of the ECDLP to compute t, a polynomial time challenger

$\mathscr{C}$ calls $\mathscr{A}_2$ as its subroutine and solves the ECDLP with a nonnegligible advantage in polynomial time.

$\mathscr{A}_2$ performs the following queries:

Setup (ID). $\mathscr{C}$ inputs the security parameter λ in the system initialization phase, picks a random number $w \in Z_q^*$ and sets $K_{\text{pub}} = w \cdot P$, and then sends the system parameter $params = \{P, p, q, E, G, h_2, h_3, h_4, K_{pub}, T_{pub}\}$ to $\mathscr{A}_2$. In this process, $\mathscr{C}$ constructs and maintains five hash lists $L_{h_{2i}}$, $L_{h_{3i}}$, $L_{h_{4i}}$, L_u, and L_{pk}, all of which are initialized to null

Create Vehicle (ID). $\mathscr{C}$ maintains a list $L_u = \{(\text{ID}, U_i, \text{vpk}_i, \text{ppk}_i, \text{vsk}_i, h_{2i})\}$. When $\mathscr{A}_2$ makes a query on ID, and if ID is in L_u, then $\mathscr{C}$ sends $(\text{ID}, U_i, \text{vpk}_i, \text{ppk}_i, \text{vsk}_i, h_2)$ to $\mathscr{A}_2$. Otherwise, if $\text{ID} = \text{ID}_\tau$, $\mathscr{C}$ selects two random numbers $x, y \in Z_q^*$ and then calculates $U_i = xP$, $\text{vpk}_i = Q$, $\text{ppk}_i = x + w \cdot h_{2i}$, $\text{vsk}_i = \perp$, and $h_{2i} = h_2(\text{ID} \| K_{\text{pub}} \| \text{params}) \longleftarrow -yx^{-1} (\text{mod } q)$. If $\text{ID} \neq \text{ID}_\tau$, $\mathscr{C}$ selects three random numbers $x, y, z \in Z_q^*$ and then calculates $U_i = x^{-1}P$, $\text{vpk}_i = (y - x^{-1})P$, $\text{ppk}_i = x^{-1} + w \cdot h_{2i}$, $\text{vsk}_i = y$, and $h_{2i} = h_2(\text{ID} \| K_{\text{pub}} \| \text{params}) \longleftarrow -yx^{-1} (\text{mod } q)$. Finally, $\mathscr{C}$ sends $(\text{ID}, U_i, \text{vpk}_i, \text{ppk}_i, \text{vsk}_i, h_2)$ to $\mathscr{A}_2$ and inserts $(\text{ID}, K_{\text{pub}}, \text{params}, h_{2i})$ into L_{h_2}

h_{2i} Query. $\mathscr{C}$ maintains a list $L_{h_2} = \{(\text{ID}, K_{\text{pub}}, \text{params}, h_{2i})\}$. When $\mathscr{C}$ receives a query $(\text{ID}, K_{\text{pub}}, \text{params})$ from $\mathscr{A}_2$, if ID is already in L_{h_2}, it sends h_{2i} to $\mathscr{A}_2$. Otherwise, $\mathscr{C}$ executes Create Vehicle (ID) to calculate $h_{2i} = h_2(\text{ID} \| K_{\text{pub}} \| \text{params})$ and then sends it to $\mathscr{A}_2$

h_{3i} Query. $\mathscr{C}$ maintains a list $L_{h_{3i}} = \{(\text{ID}, \text{params}, \nabla, h_{3i})\}$. When $\mathscr{C}$ receives a query $(\text{ID}, \text{params}, \nabla)$ from $\mathscr{A}_2$, if $L_{h_{3i}}$ contains $(\text{ID}, \text{params}, \nabla, h_{3i})$, then $\mathscr{C}$ sends h_{3i} to $\mathscr{A}_2$. Otherwise, $\mathscr{C}$ chooses a random number $h_{3i} \in Z_q^*$, calculates $h_{3i} = h_3(\text{ID} \| \nabla \| \text{params})$, sends h_{3i} to $\mathscr{A}_2$, and inserts $(\text{ID}, \text{params}, \nabla, h_{3i})$ into $L_{h_{3i}}$

h_{4i} Query. $\mathscr{C}$ maintains a list $L_{h_{4i}} = (\text{ID}, M_i, vpk_i, R_i, \nabla, t_i, h_{4i})$. When $\mathscr{A}_2$ makes a query on ID, and if ID is in $L_{h_{4i}}$, then $\mathscr{C}$ sends h_{4i} to $\mathscr{A}_2$. Otherwise, $\mathscr{C}$ selects a random number $h_{4i} \in Z_q^*$, calculates $h_{4i} = h_4(M_i \| \text{ID} \| vpk_i \| R_i \| \nabla \| t_i)$, sends h_{4i} to $\mathscr{A}_2$, and inserts $(\text{ID}, M_i, vpk_i, R_i, \nabla, t_i, h_{4i})$ into $L_{h_{4i}}$.

Partial Private Key (ID). When $\mathscr{C}$ receives a partial private key query from $\mathscr{A}_2$ for ID, $\mathscr{C}$ checks whether ID is already in L_u or not. If ID is in L_u, $\mathscr{C}$ sends ppk_i to $\mathscr{A}_2$. Otherwise, $\mathscr{C}$ runs Create Vehicle (ID) to obtain ppk_i and sends it to $\mathscr{A}_2$.

Public Key (ID). When $\mathscr{C}$ receives a public key query from $\mathscr{A}_2$ for ID, $\mathscr{C}$ checks whether ID exists in L_u or not. If it exists, $\mathscr{C}$ sends $\text{PK}_i = \{(U_i, \text{vpk}_i)\}$ to $\mathscr{A}_2$. Otherwise, $\mathscr{C}$ executes Create Vehicle (ID) to obtain the tuple (U_i, vpk_i) and sends it to $\mathscr{A}_2$

Secret Key (ID). When $\mathscr{C}$ receives a private key query from $\mathscr{A}_2$ for ID, if $\text{ID} \neq \text{ID}_\tau$, $\mathscr{C}$ checks whether L_u contains ID or not. If ID is in L_u, $\mathscr{C}$ sends vsk_i to $\mathscr{A}_2$. Otherwise, $\mathscr{C}$ executes Create Vehicle (ID) to obtain vsk_i and sends it to $\mathscr{A}_2$. In addition, if $\text{ID} = \text{ID}_\tau$, $\mathscr{C}$ stops the game.

Sign $(\text{ID}, \text{M}_\text{i})$. When $\mathscr{A}_2$ makes a signature query with (ID, M_i), $\mathscr{C}$ performs the following steps:

(1) If ID is in L_{pk}, $\mathscr{C}$ picks three random numbers $x, y, z \in Z_q^*$ such that $S_i = x$, $h_{4i} = h_4(M_i \| \text{ID} \| vpk_i \| R_i \| \nabla \| t_i) \longleftarrow x \bmod q$ and $R_i = x \cdot P + yK_{\text{pub}} - \text{xvpk}_i$, then inserts $(\text{ID}, M_i, \text{vpk}_i, R_i, \nabla, t_i, h_{4i})$ to $L_{h_{4i}}$, and finally sends the signature $\sigma_i = (R_i, S_i)$ to $\mathscr{A}_2$

(2) If ID is not in L_{pk}, because $\mathscr{C}$ knows the private key of the vehicle with ID, $\mathscr{C}$ acts like the procedure of the scheme

Forgery. In the end, $\mathscr{A}_2$ outputs a forged but valid certificateless aggregate signature $\sigma = (R, S)$ on the tuple (ID, M_i) which passes the signature verification phase. If $\text{ID} \neq \text{ID}_\tau$, $\mathscr{C}$ fails and stops

According to the forking lemma [35], $\mathscr{A}_2$ can obtain another valid signature $\sigma^* = (R^*, S^*)$ by simulating the game again with the same random tape but different choice of h_{4i} in polynomial time. We can get t according to the following equation; that is, $\mathscr{C}$ successfully solves the ECDLP. Then, we have

$$\begin{aligned} S \cdot \text{P} &= R + C + \sum_{i=1}^{n} v_i h_{4i} h_{2i} \cdot K_{\text{pub}} \\ &= R + \sum_{i=1}^{n} v_i h_{4i} \text{vpk}_i + \sum_{i=1}^{n} v_i h_{4i} h_{2i} \cdot K_{\text{pub}} \\ &= R + \left(\sum_{i=1}^{n} v_i h_{4i} \right) \cdot t \cdot P + \sum_{i=1}^{n} v_i h_{4i} h_{2i} \cdot K_{\text{pub}}, \end{aligned}$$

$$\begin{aligned} S^* \cdot \text{P} &= R + C^* + \sum_{i=1}^{n} v_i h_{4i} h_{2i} \cdot K_{\text{pub}} \\ &= R + \sum_{i=1}^{n} v_i {h_{4i}}^* C_i + \sum_{i=1}^{n} v_i h_{4i} h_{2i} \cdot K_{\text{pub}} \\ &= R + \left(\sum_{i=1}^{n} v_i {h_{4i}}^* \right) \cdot t \cdot P + \sum_{i=1}^{n} v_i h_{4i} h_{2i} \cdot K_{\text{pub}}. \end{aligned} \tag{13}$$

From these two equations, we can derive the value t by

$$\begin{aligned} (S - S^*) \cdot P &= \left(R + \left(\sum_{i=1}^{n} v_i h_{4i} \right) \cdot t \cdot P + \sum_{i=1}^{n} v_i h_{4i} h_{2i} \cdot K_{\text{pub}} \right) \\ &\quad - \left(R + \left(\sum_{i=1}^{n} v_i {h_{4i}}^* \right) \cdot t \cdot P + \sum_{i=1}^{n} v_i h_{4i} h_{2i} \cdot K_{\text{pub}} \right) \\ &\Rightarrow (S - S^*) \cdot P = v_i (h_{4i} - {h_{4i}}^*) \cdot t \cdot P \\ &\Rightarrow (S - S^*) = v_i (h_{4i} - {h_{4i}}^*) \cdot t \\ &\Rightarrow t = \frac{(S - S^*)}{v_i (h_{4i} - {h_{4i}}^*)}. \end{aligned} \tag{14}$$

Probabilistic Analysis. We analyse the advantages of $\mathscr{C}$ successfully obtaining t from $(P, Q = t \cdot P)$ to solve the ECDLP

In the process of executing Create Vehicle (ID) query, the random oracle assignment $h_{4i} = h_4(M_i \| \mathrm{ID} \| \mathrm{vpk}_i \| R_i \| \nabla \| t_i)$ causes inconsistency, which happens with probability at most q_h/q. Therefore, the probability of successful simulation of q_c times is at least $(1 - q_h/q)^{q_c} \geq 1 - q_h q_c/q$. And, the probability of successful simulation q_h times is at least $(1 - q_h/q)^{q_h} \geq 1 - {q_h}^2/q$. In addition, $\mathrm{ID} = \mathrm{ID}_\tau$ has a probability of $1/q_c$. Therefore, the overall probability of successful simulation is $\Pr[\mathrm{Succ}(\mathscr{A}_2)] \geq (1 - q_h q_c/q)(1 - {q_h}^2/q)(1/q_c)\varepsilon$. It should be noted here that the time complexity $t + O(q_c + \mathrm{q}_s)S$ of $\mathscr{C}$ is determined by the exponentiations executed in the Create Vehicle (ID) and Sign (ID, M_i) queries, where S is the time of a scalar multiplication operation.

$\mathscr{C}$ can successfully obtain t from $(P, Q = t \cdot P)$ with an advantage $(1 - q_h q_c/q)(1 - {q_h}^2/q)(1/q_c)\varepsilon$, where the time complexity of algorithm is $t + O(q_c + \mathrm{q}_s)S$, which contradicts ECDLP assumption. Therefore, the proposed scheme can resist the forgery attack of type 2 adversary $\mathscr{A}_2$ under the ROM. □

5.3. Security Analysis. Section 3.6 has given the security and privacy requirements that 5G-enabled vehicular networks. In this section, we analyse the proposed CLAS scheme to meet the above security requirements according to Theorems 1 and 2.

(i) Message integrity and authentication: according to Theorems 1 and 2, it is known that any polynomial adversary has no ability to forge a valid message. When a message msg is received from a vehicle, RSUs check the validity and integrity of the message by verifying the equation $S_i \cdot P = R_i + h_{4i}(\mathrm{vpk}_i + h_{2i} K_{\mathrm{pub}})$ to prevent the message from being tampered with and forged by malicious vehicles. As a result, no malicious adversary can construct $h_{2i} = h_2(\mathrm{PID}_i \| K_{\mathrm{pub}} \| \mathrm{params})$ and $h_{4\mathrm{i}} = h_4(M_i \| \mathrm{PID}_i \| \mathrm{vpk}_i \| R_i \| \nabla \| t_i)$ to forge a signature $\sigma_i = (R_i, S_i)$. Therefore, our proposed scheme meets the security requirements for message integrity and authentication

(ii) Privacy preserving: in 5G-enabled vehicular networks, each vehicle adopts pseudonym technology to hide its real identity RID_i to communicate, and the pseudo identity of any vehicle involves the TRA's master private key s and a random number. If an attacker wants to know the real identity of a vehicle, it can only do so through a pseudo identity. However, when given T_{pub}, $\mathrm{PID}_{i,1} = \xi_i \cdot P$, it is hard to calculate $\xi_i \cdot P \cdot T_{\mathrm{pub}}$. Besides, TRA keeps the master private key s. Therefore, our proposed scheme can meet the requirements of privacy preserving

(iii) Traceability and revocability: in our scheme, TRA can trace the identity of malicious vehicle by getting its real identity RID_i. And only TRA can extract the real identity RID_i from PID_i by its master private key through the equation $\mathrm{RID}_i = A_i \oplus h_1(\xi_i \cdot \mathrm{PID}_{i,1} \| T_{\mathrm{pub}}) \cdot \xi_i \cdot T_{\mathrm{pub}}$. When a malicious action takes place, TRA can effectively trace and revoke the malicious vehicle and insert relevant information of the malicious vehicle into RPID-Cuckoo filter for revocation. Therefore, our proposed scheme can meet the requirements of traceability and revocability

(i) Unlinkability: in our scheme, a different pseudo identity is used for each message sent by the vehicle, and the corresponding private key is used to sign the message, in which the number ξ_i in the pseudo identity is random. Moreover, there is no relationship between the new pseudo identity and the old pseudo identity for each vehicle. Therefore, the attacker cannot link two messages to the same vehicle, so our scheme meets the requirement of unlinkability

(ii) Resistance to replay attacks: in our scheme, the timestamp t_i is used to resist replay attacks. After receiving the message and signature, the receiver checks the freshness of the timestamp and whether the validity of the pseudo identity has expired. If the valid time is exceeded, the message is discarded. Therefore, our scheme is resistant to replay attacks

(iii) Resistance to modification attack: if the attacker modifies the message and sends it to others, the verifier can easily determine the message has been modified by verifying equation $S_i \cdot P = R_i + h_{4i}(\mathrm{vpk}_i + h_{2i} K_{\mathrm{pub}})$. Therefore, our scheme is resistant to modification attacks

(iv) Resistance to impersonation attack: according to Theorems 1 and 2, it is proved that our scheme is existentially unforgeable against an adaptive chosen message attack under the ROM. The adversary has not the ability to launch an impersonation attack by forging a valid signature. It is easy to determine if there is an impersonation attack by verifying the equation $S_i \cdot P = R_i + h_{4i}(\mathrm{vpk}_i + h_{2i} K_{\mathrm{pub}})$. Therefore, our scheme is resistant to impersonation attacks

(v) Resistance to information injection attack: in our scheme, the value of equation $S \cdot P = R + C + \sum_{i=1}^{n} \nu_i h_{4i} h_{2i} K_{\mathrm{pub}}$ can be changed by using a random vector

Table 3: Execution time of cryptographic operations.

Notations	Description	Run time (ms)
$T_{\mathrm{pa\_bp}}$	Bilinear pairing point addition	0.0071
$T_{\mathrm{sm\_bp}}$	Bilinear pairing scalar multiplication	1.7090
T_{bp}	Bilinear pairing	4.2110
$T_{\mathrm{sm\_ecc}}$	Elliptic curve scalar multiplication	0.4420
T_{mtp}	Map to point hash function	4.4060
$T_{\mathrm{pa\_ecc}}$	Elliptic curve point addition	0.0018
T_h	One-way hash function	0.0001

Table 4: Analysis of the computation overhead for related schemes.

Scheme	Sign	Verify	n signature verify	Without pairing	Signature scheme supports
Ali et al. [16]	$T_{sm\_bp} = 1.709ms$	$T_{sm\_bp} + T_{pa\_bp} + T_{bp} = 5.9271ms$	$nT_{sm\_bp} + nT_{pa\_bp} + T_{bp} = 1.7161n + 4.211ms$	No	Batch and aggregation
Ren et al.[15]	$2T_{sm\_bp} = 3.418ms$	$T_{pa\_bp} + 2T_{bp} = 8.4291ms$	$nT_{pa\_bp} + 2T_{bp} = 0.0071n + 8.422ms$	No	Batch and aggregation
Zhong et al.[29]	$T_{mtp} + 2T_{pa\_bp} + 4T_{sm\_bp} =$ 11.2562 ms	$2T_{mtp} + 2T_{sm\_bp} + T_{pa\_bp} + 3T_{bp} = 24.8701ms$	$nT_{mtp} + 2nT_{sm\_bp} + (2n-1)T_{pa\_bp} + 3T_{bp} = 7.8382n + 12.6259ms$	No	Aggregation
Xu et al.[37]	$T_{mtp} + T_{pa\_bp} + 3T_{sm\_bp} = 9.5401ms$	$2T_{mtp} + 2T_{sm\_bp} + T_{pa\_bp} + 3T_{bp} = 24.8701ms$	$(n+1)T_{mtp} + 2nT_{sm\_bp} + (3n-2)T_{pa\_bp} + 3T_{bp} = 7.8453\text{n} + 17.0248\text{ms}$	No	Aggregation
Gayathri et al.[36]	$2T_{sm\_ecc} = 0.884ms$	$5T_{sm\_ecc} + 3T_{pa\_ecc} = 2.2154ms$	$(n+2)T_{sm\_ecc} + 2nT_{pa\_ecc} = 0.4456n + 0.884ms$	Yes	Aggregation
Gayathri et al.[11]	$2T_{sm\_ecc} = 0.884ms$	$5T_{sm\_ecc} + 3T_{pa\_ecc} = 2.2154ms$	$5nT_{sm\_ecc} + (5n-2)T_{pa\_ecc} = 2.219n - 0.0036ms$	Yes	Batch verification
Ming and Cheng [38]	$3T_{sm\_ecc} = 1.326ms$	$4T_{sm\_ecc} + 3T_{pa\_ecc} = 1.7734ms$	$2(n+1)T_{sm\_ecc} + 3nT_{pa\_ecc} = 0.8897n + 0.884ms$	Yes	Batch verification
The proposed scheme	$T_{sm\_ecc} = 0.442ms$	$3T_{sm\_ecc} + 2T_{pa\_ecc} = 1.3296ms$	$2T_{sm\_ecc} + 2T_{pa\_ecc} = 0.8876ms$	Yes	Aggregation

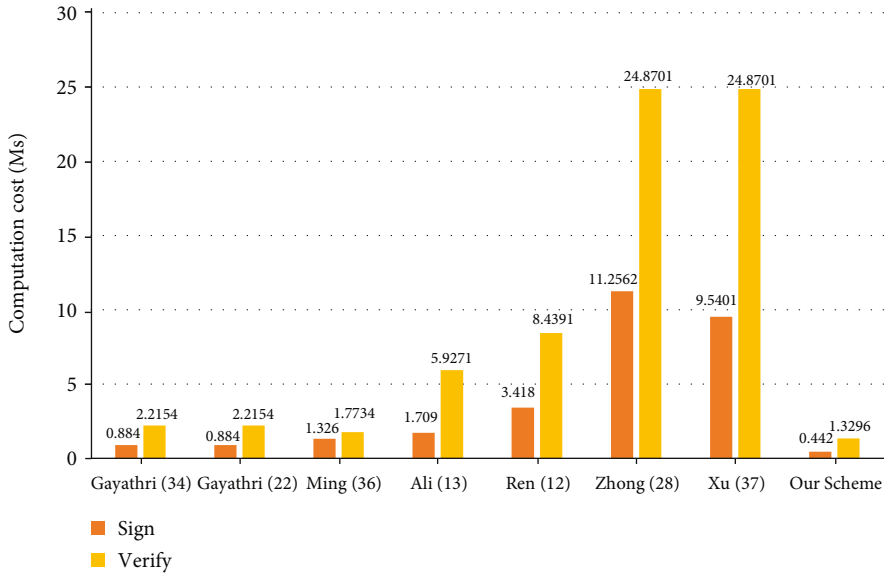

Figure 5: Computation overhead of the individual signature generation and individual signature verification.

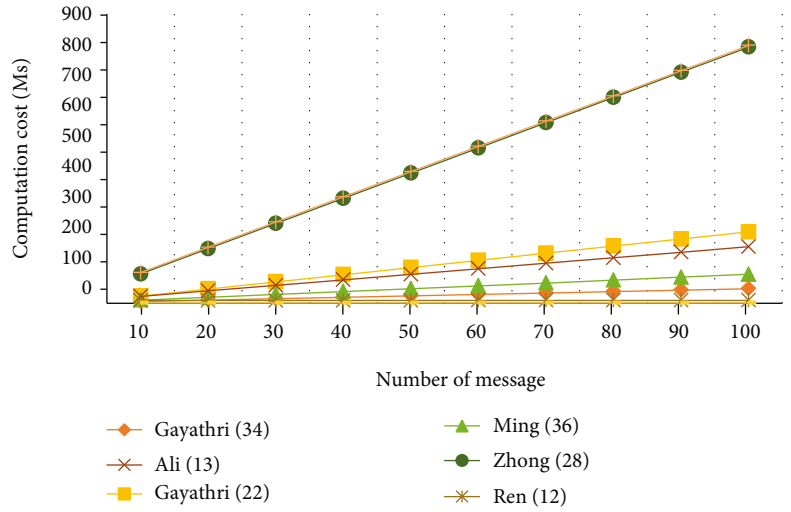

Figure 6: Aggregate verification time versus number of signatures.

Table 5: Comparison in percentage.

Scheme	Sign	Verify
Ali et al. [16]	74.13%	77.56%
Ren et al. [15]	87.06%	84.22%
Zhong et al. [29]	96.07%	94.65%
Xu et al. [37]	95.36%	94.65%
Gayathri et al. [36]	50%	39.39%
Gayathri et al. [11]	50%	39.39%
Ming and Cheng [38]	66.67%	24.80%

v in the aggregation verification phase. Therefore, our scheme is resistant to information injection attacks

Table 6: Variables used for communication overhead analysis.

Symbol	Description	Length(bytes)
$\|G_1\|$	The size of elements in group G_1	128 bytes
$\|G\|$	The size of elements in group G	40 bytes
$\|q\|$	The size of the elements in Z_q^*	20 bytes

6. Performance Analysis

In this section, we compare our proposed scheme with some schemes proposed [11, 15, 16, 29, 36–38] in order to evaluate its performance from the perspective of computational overhead and communication overhead.

Table 7: Communication overhead comparison.

Scheme	Single signature	Aggregate signature
Ali et al. [16]	$\|G_1\| = 128$ bytes	$\|G_1\| = 128$ bytes
Zhong et al. [29]	$2\|G_1\| = 256$ bytes	$2\|G_1\| = 256$ bytes
Xu et al. [37]	$2\|G_1\| = 256$ bytes	$(n+1)\|G_1\| = 128n + 128$ bytes
Our scheme	$\|G\| + \left\|Z_q^*\right\| = 60$ bytes	$2\|G\| + \left\|Z_q^*\right\| = 100$ bytes

6.1. Computational Overhead Analysis. We adopt the same evaluation method as He et al. [10] to analyse the computational overhead of the proposed scheme. The basic configuration is 3.40GHz clock frequency, 4GB of running memory, an Intel i7-4770 processor, and MIRACL library running on the Windows 7 operating system. The MIRACL library is a well-known cryptographic library. The bilinear pairing on the security level of 80 bits is constructed as $\overline{e} : G_1 \times G_2 \longrightarrow G_2$, where G_1 is an additive group of order $\widehat{q}$ defined on a super-singular elliptic curve $\bar{E} : y^2 = x^3 \bmod \widehat{p}$ with embedding degree of 2. Here, we consider $\widehat{p}$ as a 512-bit prime number and $\widehat{q}$ as a 160-bit solinas prime number. The ECC on the security level of 80 bits is created on the nonsingular elliptic curve $E : y^2 = x^3 + ax + b \bmod p$ as follows: G is an additive group of order q, where p, q are 160-bit primes and $a, b \in z_p^*$. The execution time of the basic cryptographic operations in the scheme is shown in Table 3.

We compare the computational overhead of our scheme with these schemes [11, 15, 16, 29, 36–38] in three phases: individual signature generation phase, individual signature verification phase, and aggregate signature verification phase. These schemes [15, 16, 29, 37] are based on bilinear pair construction, and these schemes [11, 36, 38] and our scheme are constructed using ECC. Since the general one-way hash operation T_h only incurs very little overhead in the signing and verification process, we no longer count the running time of this operation. The specific analysis of the computation overhead is shown in Table 4.

The computation overhead of the individual signature generation phase and individual signature verification phase are shown in Figure 5. It is seen that a signature needs to consist of bilinear pairing based four scalar multiplications and two scalar point additions. It also needs a map to point hash function operation in Zhong et al.'s scheme [29]. Therefore, the computational cost of the signature generation phase of the scheme is $T_{\text{mtp}} + 2T_{\text{pa\_bp}} + 4T_{\text{sm\_bp}} = 11.2562$ ms. In the signature verification phase, operations performed include three bilinear pairing operations, two scalar multiplication operations based on bilinear pairing, one scalar point addition operation based bilinear pairing, and two map to point hash operations. Thus, the computation cost is $2T_{\text{mtp}} + 2T_{\text{sm\_bp}} + T_{\text{pa\_bp}} + 3T_{\text{bp}} = 24.8701$ ms. The aggregate signature verification phase of Zhong et al.'s [29] scheme requires three bilinear pairing operations, $2n$ scalar multiplication operations based on bilinear pairing, $(2n-1)$ scalar point addition operations based on bilinear pairing and n map to point hash operations, where n is the number of sensing information transmitted by the sender within a period of time. In this phase, the computation cost is $nT_{\text{mtp}} + 2nT_{\text{sm\_bp}} + (2n-1)T_{\text{pa\_bp}} + 3T_{\text{bp}} = 7.8382n + 12.6259$ ms.

In the proposed scheme, the signature generation phase requires one scalar multiplication operation based ECC and one scalar point addition operation based ECC, whose computational cost is $T_{\text{sm\_ecc}} = 0.442$ ms in this phase. In the signature verification phase, operations performed include three scalar multiplication operations based ECC and two scalar point addition operations based ECC. The computational cost in this phase is $3T_{\text{sm\_ecc}} + 2T_{\text{pa\_ecc}} = 1.3296$ ms. In the aggregate signature verification phase, operations performed include two scalar multiplication operations based ECC and two scalar point addition operations based ECC. In the aggregate signature verification phase of this scheme, the computation cost is $2T_{\text{sm\_ecc}} + 2T_{\text{pa\_ecc}} = 0.8876$ ms.

Computational cost in these schemes [11, 15, 16, 36–38] can be analysed by using similar analysis methods described. The aggregate verification time relating to the number of signatures is given as Figure 6. It can be observed that the schemes of Ren et al. [15], Ali et al. [16], and Gayathri et al. [37] adopted scalar multiplication operation and scalar point addition operation-based bilinear pairing and map to point hash operations. It is well known that bilinear pairing operations are time-consuming. In addition, despite the application of ECC, the schemes of Gayathri et al. [11, 36] and Ming and Cheng [38] failed to outperform the proposed scheme. For example, in the n signature verification phase of the schemes in [11, 36, 38], the computation cost requires $5nT_{\text{sm\_ecc}} + (5n-2)T_{\text{pa\_ecc}} = 2.219n - 0.0036$ ms, $(n+2)T_{\text{sm\_ecc}} + 2nT_{\text{pa\_ecc}} = 0.4456n + 0.884$ ms, and $2(n+1)T_{\text{sm\_ecc}} + 3nT_{\text{pa\_ecc}} = 0.8897n + 0.884$ ms, respectively. However, the proposed scheme only needs $2T_{\text{sm\_ecc}} + T_{\text{pa\_ecc}} = 0.8858$ ms to verify n signatures. Compared with the scheme of Zhong et al. [29], the performance efficiency of the proposed scheme is about $((11.2562 - 0.442)/11.2562) \times 100\% \approx 96.07\%$ and $((24.8701 - 1.3296)/24.8701) \times 100\% \approx 94.65\%$, respectively. Similarly, compared with the scheme of Xu et al. [37], the proposed scheme improves signature generation and signature verification by approximately 95.36% and 94.65%. Our scheme improves signature generation and signature verification by approximately 87.06% and 84.22% over Ren et al.'s [15] scheme. Compared with Ali et al.'s [16] scheme, our scheme improves signature generation and signature verification by about 74.13% and 77.56%. Our scheme improves signature generation and signature verification by approximately 66.67% and 24.80% over Ming et al.'s [38]

scheme. Compared with Gayathri et al.'s [11, 36] scheme, our scheme improves signature generation and signature verification by about 50% and 40%. In addition, with respect to total computation overhead, the proposed scheme increases performances by 98.55%, 87.32%, 94.52%, 98.81%, and 97.13%, respectively, compared to the CLAS schemes [11, 15, 16, 36, 38]. It can be observed from the above results that the proposed scheme outperforms other related works. The comparison in percentage is shown in Table 5.

6.2. Communication Overhead Analysis. Driven by 5G, the IoVs need to support the interconnection between smart devices and systems. It will provide access to users everywhere at any time and bring high-intensity data transmission pressure to vehicular networks. Therefore, it is crucial to reduce communication overhead. For fairness, we use the variables in Table 6 to simulate the equivalent security levels of the schemes. On the 80-bit security level, let the sizes of the elements in G and G_1 be 40 bytes and 128 bytes, where G and G_1 are an additive group on a nonsingular elliptic curve and an additive group on a super singular curve, respectively. The signature sizes of the schemes [15, 29, 36, 37] are compared with that of the proposed scheme as shown in Table 7. The aggregate signature generated by the proposed scheme consists of one Z_q^* element and one G element based on elliptic curve; therefore, it only takes 100 bytes to send the aggregate signature. It should be noted that the size of the aggregate signature bears no relation with the number of messages in the scheme of Ren et al. [15], Zhong et al. [29], Gayathri et al. [36], and our scheme. Although these schemes have lower communication cost, the length of Ali et al.'s scheme [16] and Zhong et al.'s scheme [29] are 128 bytes and 256 bytes, respectively, which far exceeds communication cost required by our scheme to send an aggregate signature. Due to the proposed scheme has lower transmission cost than other schemes, it can meet the low delay requirements for 5G-enabled vehicular networks.

7. Conclusion

This paper proposes a lightweight CLAS scheme with revocation mechanism for 5G-enabled vehicular networks. The proposed scheme uses binary search to identify invalid signatures and introduces a cuckoo filter to revoke malicious users to prevent reattack. The security analysis shows that the proposed scheme is secure under ROM and meets all security and privacy requirements. Moreover, our CLAS scheme uses ECC to construct the specific algorithm, which does not need the computationally complex bilinear pairing operations and map to the point hash functions, thus improving the computational efficiency. Through experimental comparison with other related schemes, our CLAS scheme has significant advantages in computational overhead and communication overhead.

Data Availability

The proposed scheme and its analysis need only theoretical and experimental support. There is no additional data set to be provided in this paper.

Conflicts of Interest

The authors declare that they are no conflicts of interest in this paper.

Acknowledgments

This work was sponsored in part by the National Natural Science Foundation of China under Grants 61941113, 61971033, and 61671057; by the Henan Provincial Department of Science and Technology Project (No. 212102210408); and by the Henan Provincial Key Scientific Research Project (No. 22A520041).

References

[1] J. G. Andrews, S. Buzzi, W. Choi et al., "What will 5g be?," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[2] J. Zhang, H. Zhong, J. Cui, M. Tian, Y. Xu, and L. Liu, "Edge computing-based privacy-preserving authentication framework and protocol for 5G-enabled vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7940–7954, 2020.

[3] M. H. Eiza, N. Qiang, and S. Qi, "Secure and privacy-aware cloud-assisted video reporting service in 5G-enabled vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 7868–7881, 2016.

[4] S. Chen, J. Hu, Y. Shi, L. Zhao, and W. Li, "A vision of C-V2X: technologies, field testing, and challenges with Chinese development," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3872–3881, 2020.

[5] S. Chen, J. Hu, Y. Shi, and L. Zhao, "LTE-V: a TD-LTE-based V2X solution for future vehicular network," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 997–1005, 2016.

[6] Z. Wang, C. Guo, J. Liu et al., "Accurate and privacy-preserving task allocation for edge computing assisted mobile crowdsensing," *IEEE Transactions on Computational Social Systems*, vol. 9, pp. 120–133, 2022.

[7] M. Raya, P. Papadimitratos, and J.-P. Hubaux, "Securing vehicular communications," *IEEE Wireless Communications*, vol. 13, no. 5, pp. 8–15, 2006.

[8] J. P. Hubaux, S. Capkun, and J. Luo, "The security and privacy of smart vehicles," *IEEE Security Privacy*, vol. 2, no. 3, pp. 49–55, 2004.

[9] M. A. Al-shareeda, M. Anbar, S. Manickam, and I. H. Hasbullah, "An efficient identity-based conditional privacy-preserving authentication scheme for secure communication in a vehicular ad hoc network," *Symmetry*, vol. 12, no. 10, p. 1687, 2020.

[10] D. He, S. Zeadally, B. Xu, and X. Huang, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *IEEE Trans. Information Forensics and Security*, vol. 10, no. 12, pp. 2681–2691, 2015.

[11] N. B. Gayathri, G. Thumbur, P. V. Reddy, and M. Z. U. Rahman, "Efficient pairing-free certificateless authentication scheme with batch verification for vehicular ad-hoc networks," *IEEE Access*, vol. 6, pp. 31808–31819, 2018.

[12] M. Bayat, M. Barmshoory, M. Rahimi, and M. R. Aref, "A secure authentication scheme for VANETs with batch verification," *Wireless networks*, vol. 21, no. 5, pp. 1733–1743, 2015.

[13] J. Li and Y. Zhang, "Cryptanalysis and improvement of batch verification certificateless signature scheme for VANETs," *Wireless Personal Communications*, vol. 111, no. 2, pp. 1–15, 2019.

[14] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, *Aggregate and Verifiably Encrypted Signatures from Bilinear Maps*, Proceedings of the International Conference on theory and Applications of Cryptographic Techniques, Advances in Cryptology — EUROCRYPT 2003, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2003.

[15] Y. Ren, X. Li, S. F. Sun, X. Yuan, and X. Zhang, "Privacy-preserving batch verification signature scheme based on blockchain for vehicular ad-hoc networks," *Journal of Information Security and Applications*, vol. 58, article 102698, 2021.

[16] I. Ali, M. Gervais, E. Ahene, and F. Li, "A blockchain-based certificateless public key signature scheme for vehicle-to-infrastructure communication in VANETs," *Journal of Systems Architecture*, vol. 99, article 101636, 2019.

[17] Q. Mei, H. Xiong, J. Chen, M. Yang, S. Kumari, and M. K. Khan, "Efficient certificateless aggregate signature with conditional privacy preservation in IoV," *IEEE Systems Journal*, vol. 15, no. 1, pp. 245–256, 2021.

[18] J. Cui, X. Zhang, H. Zhong, Z. Ying, and L. Liu, "RSMA: Reputation system-based lightweight message authentication framework and protocol for 5G-enabled vehicular networks," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6417–6428, 2019.

[19] S. Taha and X. Shen, "Lightweight group authentication with dynamic vehicle-clustering for 5G-based V2X communications," in *In 2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Abu Dhabi, United Arab Emirates, 2018.

[20] A. Shamir, "Identity-based cryptosystems and signature schemes," in *In Workshop on the theory and application of cryptographic techniques, Advances in Cryptology*Springer, Berlin, Heidelberg.

[21] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *In International conference on the theory and application of cryptology and information security*, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2003.

[22] D. H. Yum and P. J. Lee, *Generic Construction of Certificateless Signature*, Proceedings of the Australasian Conference on Information Security and Privacy (ACISP) Information Security and Privacy, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2004.

[23] B. C. Hu, D. S. Wong, Z. Zhang, and X. Deng, "Key replacement attack against a generic construction of certificateless signature," in *Australasian Conference on Information Security and Privacy, Information Security and Privacy*, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2006.

[24] X. Huang, W. Susilo, Y. Mu, and F. Zhang, "On the security of certificateless signature schemes from Asiacrypt 2003," in *In International Conference on Cryptology and Network Security, Cryptology and Network Security*, Lecture Notes in Computer Science, Springer Berlin, Heidelberg, 2005.

[25] J. Cui, J. Zhang, H. Zhong, R. Shi, and Y. Xu, "An efficient certificateless aggregate signature without pairings for vehicular ad hoc networks," *Information Sciences*, vol. 451, pp. 1–15, 2018.

[26] I. A. Kamil and S. O. Ogundoyin, "An improved certificateless aggregate signature scheme without bilinear pairings for vehicular ad hoc networks," *Journal of Information Security and Applications*, vol. 44, pp. 184–200, 2019.

[27] Y. Zhao, Y. Hou, L. Wang, S. Kumari, M. K. Khan, and H. Xiong, "An efficient certificateless aggregate signature scheme for the Internet of Vehicles," *Transactions on Emerging Telecommunications Technologies*, vol. 31, pp. 1–20, 2020.

[28] G. Thumbur, G. S. Rao, P. V. Reddy, N. B. Gayathri, D. K. Reddy, and M. Padmavathamma, "Efficient and secure certificateless aggregate signature-based authentication scheme for vehicular ad hoc networks," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1908–1920, 2021.

[29] H. Zhong, S. Han, J. Cui, J. Zhang, and Y. Xu, "Privacy-preserving authentication scheme with full aggregation in VANET," *Information Sciences*, vol. 476, pp. 211–221, 2019.

[30] I. A. Kamil and S. O. Ogundoyin, "On the security of privacy preserving authentication scheme with full aggregation in vehicular ad hoc network," *Security Privacy*, vol. 3, no. 3, article e104, 2020.

[31] J. Cui, J. Chen, H. Zhong, J. Zhang, and L. Liu, "Reliable and efficient content sharing for 5G-enabled vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1247–1259, 2020.

[32] P. Wang, C. M. Chen, S. Kumari, M. Shojafar, R. Tafazolli, and Y. N. Liu, "HDMA: hybrid D2D message authentication scheme for 5G-enabled vanets," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5071–5080, 2021.

[33] B. Fan, D. G. Andersen, M. Kaminsky, and M. D. Mitzenmacher, "Cuckoo filter: practically better than bloom," in *In: Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies*, pp. 75–88, New York, NY, USA, 2014.

[34] D. He, J. Chen, and R. Zhang, "An efficient and provably-secure certificateless signature scheme without bilinear pairings," *Int. J. Commun Syst*, vol. 25, no. 11, pp. 1432–1442, 2012.

[35] P. David and S. Jacque, "Security arguments for digital signatures and blind signatures," *Journal of Cryptography*, vol. 13, no. 3, pp. 361–396, 2000.

[36] N. B. Gayathri, T. Gowri, P. R. Kumar, M. Z. U. Rahman, P. V. Reddy, and A. Lay-Ekuakille, "Efficient and secure pairing-free certificateless aggregate signature scheme for healthcare wireless medical sensor networks," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 9064–9075, 2019.

[37] Z. Xu, D. He, N. Kumar, and K.-K. R. Choo, "Efficient certificateless aggregate signature scheme for performing secure routing in VANETs," *Security and Communication Networks*, vol. 2020, Article ID 5276813, 2020.

[38] Y. Ming and H. Cheng, "Efficient certificateless conditional privacy-preserving authentication scheme in VANETs," *Mobile Information Systems*, vol. 2019, Article ID 7593138, 2019.