# An Efficient and Provably Secure ECC-Based Conditional Privacy-Preserving Authentication for Vehicle-to-Vehicle Communication in VANETs

Ikram Ali, Yong Chen, *Senior Member, IEEE*, Niamat Ullah, Rajesh Kumar, and Wen He

***Abstract*—In vehicular ad-hoc networks (VANETs), safety messages are exchanged among vehicles and between vehicles and infrastructure to ensure passengers' safety and efficiency in traffic. The source authentication as well as integrity checking of these messages are very necessary for a receiver. Based on certificateless cryptography (CLC), some state-of-the-art signature schemes have been proposed to address these. Although they fulfill the requirements of authentication and privacy, they are not efficient with respect to performance. Bilinear pairings and map-to-point hash functions are used in these schemes. These require a huge amount of time to process. The computational power and storing capacity of an on-board unit (OBU) in each vehicle are limited. Therefore, computational overhead is induced on vehicles that need to authenticate messages in areas of high traffic density. In this paper, a provably secure and efficient certificateless short signature-based conditional privacy-preserving authentication (CLSS-CPPA) scheme for V2V communication is designed. This scheme does not use bilinear parings and is based on the elliptic curve cryptosystem (ECC). In addition, instead of map-to-point hash functions, general hash functions are used. Furthermore, the CLSS-CPPA scheme supports the batch signature verification method which allows multiple signatures to be verified simultaneously and efficiently. The CLSS-CPPA scheme ensures security against type-I and type-II attackers with respect to existential unforgeability against adaptively chosen message attacks (EUF-CMA) under a hardness assumption of the elliptic curve discrete logarithm problem (ECDLP) in the random oracle model (ROM). The proposed scheme significantly improves performance in terms of computational and communication costs in comparison with state-of-the-art schemes.**

***Index Terms*—Anonymous-identity, authentication, computational cost, vehicle, certificateless cryptography.**

Manuscript received October 5, 2020; revised December 9, 2020; accepted January 5, 2021. Date of publication January 13, 2021; date of current version March 10, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61973331 and in part by the National Key Research and Development Plan Programs of China under Grant 2018YFB0106101. The review of this article was coordinated by Prof. X. Du. *(Corresponding author: Yong Chen.)*

Ikram Ali and Yong Chen are with the School of Automation Engineering and Institute of Electric Vehicle Driving System and Safety Technology, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: ikramcs2@gmail.com; ychencd@uestc.edu.cn).

Wen He is with the School of Automation Engineering and Institute of Electric Vehicle Driving System and Safety Technology, University of Electronic Science and Technology of China, Chengdu 611731, China, and also with the Intelligent Research Institute, Chongqing Changan Automobile Technology Company Ltd, Chongqing 400020, China (e-mail: hewen@changan.com.cn).

Rajesh Kumar is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: rajakumarlohano@gmail.com).

Niamat Ullah is with the Department of Computer Science, University of Buner, Khyber Pakhtunkhwa 19290, Pakistan (e-mail: niamatnaz@gmail.com).

Digital Object Identifier 10.1109/TVT.2021.3050399

## I. Introduction

THE management of traffic on roads in congested cities is a daunting task and very often results in unpleasant events such as traffic jams, accidents, wastage of fuel, time, etc. Therefore, it is a requirement of recent time to manage traffic on roads in such cities securely and efficiently. The fast growing wireless communication technologies [1] facilitate intelligent transportation systems (ITS) due to which traffic generated by thousands of vehicles is managed efficiently. ITS applications in the form of vehicular ad-hoc networks (VANETs) allow vehicles to have equipment in the form of on-board units (OBU's) to be installed on them. The OBU in each vehicle communicates safety messages to the OBUs in neighboring vehicles and infrastructure (such as the road-side unit (RSU) and the trusted authority (TA)) within a range of 300 meters through a dedicated short range communications (DSRC) system [3], [4]. The DSRC is a pioneer ITS project of the US department of transportation dedicated to VANET standardization. VANETs support two modes of communication: vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) [5]. In V2V, communicating vehicles exchange information about their speed, location, heading, traffic jams, etc with each other to avoid crashes, ease traffic congestion, and improve the driving environment. They transmit and receive omni-directional messages (up to 10 times per second to create a 360-degree alertness of other vehicles in their immediate environment) [6]. V2I creates a communication network between vehicles and infrastructure to improve safety, comfort, and efficiency. In addition, Internet services are provided through V2I to deliver audio/video applications to drivers and passengers. Based on the aforementioned advantages/services, drivers can make prompt decisions which enhance the overall driving safety and efficiency. However, irrespective of the benefits they provide, VANETs still face problems related to authentication and privacy [7]–[10]. The vehicles in VANETs communicate with each other through a wireless medium. An attacker can utilize that when launching attacks. For example, the attacker can collect sensitive messages, change them, and then spread them in VANETs to cause serious damage. Therefore, messages that are broadcast in VANETs should ensure authentication, integrity and other security requirements. In addition, it is necessary to ensure each vehicle's privacy, i.e., protect its original identity from malicious vehicles that try to get information about its identity, current position, and direction [11]–[13].

In VANETs, conditional privacy-preservation is also necessary, i.e., the TA traces the original identities of malicious vehicles in a case of misconduct. Therefore, it is crucial to consider all the aforementioned issues related to security and privacy prior to implementing VANETs.

To address the above mentioned issues, researchers have proposed many schemes. However, these schemes concentrate on authentication and integrity without giving enough attention to how efficiently this results are achieved. Vehicles in VANETs travel at high speeds and each vehicle generates signed messages that are authenticated by a receiving vehicle via a verification of signatures. The maximum amount of time allowed for a signature verification ranges from 100–300 milliseconds (ms) [3]. This is the interval in ms between each message transmitted by an RSU. For instance, if an emergency message (about an accident) is transmitted to other vehicles through the RSU via V2I, such a message is first transmitted to the concerned RSU by a vehicle that observed the event. The source and integrity of the message is authenticated and verified by the RSU respectively. The result of this verification is then broadcast to other vehicles within the RSU communication range. However, the vehicles that are outside the RSU range will not receive this message directly [14]. So they will not be able to take preventive measures in time. This barrier can be overcome if the V2V mode of communication is adapted [6]. This will enable each vehicle to communicate with the nearby vehicles directly without the involvement of the RSU. This, will improve coverage and transmission speed.

### A. Motivations and Contributions

Authentication schemes for V2V communication still face a lot of problems when it comes to their performance. Mainly, they are still deficient in areas of aggregate authentication of multiple safety messages in densely populated areas. This is mainly due to computationally demanding operations such as bilinear pairings and map-to-point hash functions. These operations incur very high computational costs and as such they greatly increase the cost of message signing and verification. For instance, the cost required for one bilinear pairing operation is almost twenty times larger than that of the cost required for one scalar multiplication operation over the elliptic curve's group [15]. Also, the amount of time required to process a map-to-point hash function greatly exceeds the amount of time required to process a general hash function. Due to the above operations, it is very difficult for a receiver vehicle to authenticate multiple messages generated from multiple vehicles during every 100-300 ms [3] sequentially. This, is due to the limited resources (computational power, storage capacity, energy, etc.). This computational overhead is what motivates us to design an efficient and provably secure CPPA scheme that is capable to authenticate multiple safety messages simultaneously for V2V communication. Our contributions are as follows:

- First, we propose a certificateless short signature-based conditional privacy-preserving authentication (CLSS-CPPA) scheme based on the elliptic curve cryptosystem (ECC) [16] without using bilinear pairing for V2V communication. Instead of map-to-point hash functions, the proposed scheme uses general one-way hash functions. In order to enable a receiver vehicle to verify multiple messages simultaneously instead of one by one, the proposed scheme supports the batch signature verification method. This speeds up the performance of V2V communication.
- Second, under the hardness hypothesis of the elliptic curve discrete logarithm problem (ECDLP), we prove security of the CLSS-CPPA scheme with respect to existential unforgeability against adaptively chosen message attacks (EUF-CMA) against type-I and type-II attackers in the ROM.
- Finally, to indicate that our CLSS-CPPA scheme has significant improvement in efficiency, we evaluate the performance of the proposed scheme in terms of its computational and communication cost in comparison with some state-of-the-art signature schemes.

The organization of the rest of the paper is as follows: We provide a detailed review of related works in Section II. Sections III and IV provide a brief introduction to the preliminary knowledge used in the development of the proposed scheme as well as the framework and security notions used in its design. A detailed description of proposed scheme is given in Section V. In Sections VI and VII, security and performance of the CLSS-CPPA scheme are analyzed comprehensively. Section VIII concludes this paper.

## II. Related Works

To ensure the safety of messages as well as the partial privacy of vehicles, a large volume of relevant signature schemes based on the public key infrastructure (PKI), identity-based cryptography (IDC), and certificateless cryptography (CLC) were proposed. Some of these schemes based on the PKI that were proposed specifically for VANETs include [17]–[22]. These schemes are constructed based on the traditional public key cryptography (PKC) which makes use of public/private keys and a certificate from a trusted authority, i.e., certificate authority (CA). In the PKI, each vehicle first gives its public key and identity to CA. After verification, the CA issues a certificate to it. The certificate provides the authenticity of the public key. In the PKI, for each vehicle, a public/private key pair is generated. The management (i.e., generation, transmission, storage, verification, and revocation) of certificates results in an overhead with respect to computational and communication cost. This causes problems for the receiver vehicle, i.e., computational overhead is increased. In addition to this, signature and certificate increase the size of the packet in transmission due to which communication overhead is increased.

In 1984, Shamir [23] introduced the IDC. It solved the problem of certificate management. The IDC does not require certificates. Instead, it uses some sort of the users identity (an email address, telephone number, etc.,) as its public key. A private key generator (PKG) is used to generate a corresponding private key of the user. Shim [24] designed an ID-based CPPA scheme for VANETs. However, the three bilinear pairings and one extra multiplication required by the scheme makes it inefficient when it comes to signature verification. This observation

is on the grounds that a bilinear pairing operation requires an immense amount of time to process. He *et al.* [25] and Lo and Tsai [26] later addressed a flaw they found in Shim's scheme [24] and proposed the ECC-based CPPA schemes based on the IDC for VANETs. In any case, the computational cost required for three scalar multiplications can even now result in a considerable overhead in signature verification. In [27] and [28], Zhang *et al.* and Bayat *et al.* addressed the problem that Lee and Lai's scheme [29] does not resist attacks (replay, and impersonation), ensure traceability, and non-repudiation. They also modified the anonymous-identity and signature generation phase in [29] to design CPPA schemes based on the IDC. With regard to computational cost during a signature verification, these schemes are not efficient because of the three bilinear pairing operations. Also, these schemes do not ensure security against modification attacks in line with Liu *et al.* [30]. Cui *et al.* [31] in 2017, presented an ECC-based ID-CPPA scheme for VANETs. Their scheme uses a cuckoo filter and binary search methods. Recently, we proposed a bilinear pairing-based CPPA scheme for V2I communication [32]. However, in our scheme only one bilinear pairing operation can still produce delay in the signature verification. Very recently, we designed an efficient ECC-based CPPA scheme in [14] for V2V communication. The main advantage of our scheme is that it is provably secure and is efficient with respect to the computational complexity. However, within the IDC-based schemes the key escrow problem is the main common setback, i.e., the PKG can use a user's private key and forge a signature.

In order to resolve this, the CLC mechanism was introduced by Al-Riyami and Paterson in 2003 [33]. First a key generation center (KGC) generates a partial private key and transmits it to a user in the CLC. Using the partial private key and a random secret value chosen by himself, a full private key is then generated by the user. A CLC-based signature scheme that uses a signature aggregation method to authenticate messages in ad-hoc networks was proposed by Xiong *et al.* [34]. However, the three bilinear pairings and two scalar multiplications can delay a signature verification. In addition, this scheme does not ensure security against type-II attacks according to He *et al.* [35]. This is because an attacker of type-II can forge a valid signature on any message. In [36], Malip *et al.* proposed a CPPA scheme based on the CLC using bilinear pairings and supports aggregate signature verification in VANETs. This scheme is not fast due to the use of time-consuming operations in message signing and verification. In 2015, Malhi and Batra [37] presented an efficient CLC-based scheme that supports signature aggregation for VANETs. A provably secure CLC-based CPPA scheme that uses bilinear pairing for V2V communication was proposed by Horng *et al.* [38]. Both batch and aggregate signature verifications are supported by their scheme. However, a large number of time-consuming operations used during signature verification renders the authentication process inefficient. Li *et al.* [39] proposed a CLS scheme by executing cryptanalysis on Horng *et al.*'s scheme [38]. He analyzed the weakness to malicious-but-passive KGC attacks in their scheme and proposed a secure CLS scheme based on aggregation. A short signature scheme based on the CLC by utilizing bilinear pairing was designed by Tsai [40].

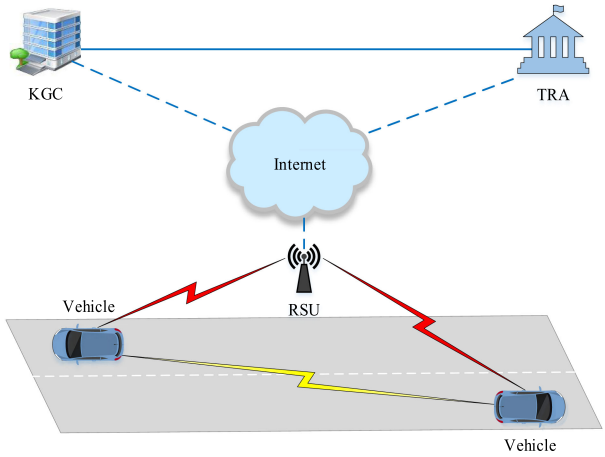


Fig. 1. Proposed system model.

The single bilinear pairing operation makes this scheme efficient with respect to signature verification. It should be noted that, this scheme does not ensure privacy of identity. In 2018, a scheme that uses bilinear pairings and supports signature aggregation verification was proposed by Kumar *et al.* [41]. This was a CLC-based CPPA scheme. An ECC-based CPPA scheme based on the CLC for V2I communication was proposed by Cui *et al.* [9]. The aggregate signature verification reduces verification delay and improves bandwidth. However, according to Kamil and Ogundoyin [42], Cui *et al.*'s scheme [9] can not provide existential unforgeability against type-II attacks in the random oracle model (ROM). Very recently, we designed a CPPA scheme based on blockchain for V2V communication [10]. This scheme supports the batch signature verification as well as aggregate signature verification. However, the single bilinear pairing operation can still delay the signature verification. To improve the efficiency in signature generation and verification, a CLSS-CPPA scheme without using a bilinear paring has been proposed in this paper.

## III. Preliminaries

For the design of our CLSS-CPPA scheme, we provide a brief overview on system model, security requirements, the ECC, and computational hard problems in this section.

### A. System Model

Our scheme's system model is based on four entities: the OBU, the RSU, and two TAs (i.e., tracing authority (TRA) and KGC) as shown in Fig. 1. Below we describe each of these entities in detail:

1) *OBU:* An OBU installed in each vehicle receives messages from some source (i.e., vehicle or sensor), verifies them, and transmits them to other vehicles via the DSRC system [3]. The secret credentials for a message signing and verification are kept secret in a tamper-proof device (TPD) within the OBU. Each OBU in VANETs contains a clock, which is securely resynchronized within an RSU's communication range. In addition to this, it equipped with a global positioning system (GPS) and graphical user

interface (GUI) to provide services about location and interaction with drivers, respectively. Its computational power and storing capacity are less than that of the RSU.

2) *RSU:* It is a base-station fixed along a roadside to work as intermediate entity among vehicles, TRA, and KGC. It warn incoming vehicles of the probable threat and informs them to decrease the speed and change the route. Its computational power and storing capacity are less than that of the TRA and KGC. It receives messages, authenticates them, and then broadcasts them within its communication range or forwards the verified messages to another entity (i.e., application server) through a secure transmission channel (i.e., wired TLS protocol).
3) *TAs (TRA and KGC):* These are the upper layer authorities having the responsibility of managing the whole VANETs system. The TRA first registers the RSU and vehicles and then generates anonymous-identities to preserve privacy of each vehicle. It has the authority to trace the original identity of a misbehaving vehicle and can revoke its registration. While the KGC generates partial private keys and assigns them to vehicles. In our proposed model, we consider that the TRA and KGC are trusted in VANETs.

### B. Security Requirements

The proposed CLSS-CPPA scheme in V2V communication should fulfill the following security requirements.

1) Message authentication and integrity: A receiver vehicle should authenticate the source of a message and that the message should not be altered in transmission.
2) Privacy (identity-anonymity): A vehicle's original identity should be anonymous whilst it communicates with other vehicles.
3) Non-repudiation: A sender vehicle must not be able to refuse the messages which they have sent.
4) Conditional traceability: Only the TRA should obtain a vehicle's original identity from the concerned anonymous-identity when a disputed message is received.
5) Unlinkability: Malicious vehicles, RSUs, and third parties should not know that two or more messages are sent from the same vehicle. There should be no way to link them.
6) Resistance against attacks: The CLSS-CPPA scheme should ensure resistance against modification attacks, impersonation attacks, man-in-the-middle attacks, and replay attacks.

### C. Elliptic Curve Cryptosystem (ECC)

For the first time in 1984, the elliptic curve was used in cryptography by Miller [16]. After that Kobilitz designed the ECC with a discrete logarithm problem based on the elliptic curve. Later, the ECC became widely used in encryption protocols and other crypto-systems. The basics of the ECC are as follows:

Consider a finite field $\mathbb{F}_p$, which has prime order $p$. $\mathbb{F}_p$ is based on an elliptic curve $E$. $E$ is a non-singular elliptic curve that uses an equation $y^2 \equiv (x^3 + ax + b) \bmod p$, for $4a^3 + 27b^2 \neq 0$ and $(a, b) \in \mathbb{F}_p$. On $E$, suppose a point $\mathcal{O}$ at infinity and some other points, i.e., $P$, $R$, and $W$ make an additive group $\mathbb{G}$ with a prime order $q$. A point $P$ generates the group $\mathbb{G}$. In the ECC, the group $\mathbb{G}$ has properties, which are as follows:

- Point addition: Suppose two random points $P$ and $W$ on $E$ such that $(P, W) \in \mathbb{G}$, where the group $\mathbb{G}$ with order $q$ is generated by the point $P$. $R = P + W$ if $P \neq W$, $R = 2P$ if $P = W$, and $P + W = \mathcal{O}$ if $P = -W$.
- Scalar multiplication: Suppose the scalar multiplication or point multiplication on $E$ is given as $lW = \underbrace{W + W + \cdots + W}_{l}$ for $l$ times, where $l \in \mathbb{Z}_q^*$ and $l > 0$.

### D. Computational Hard Problems

The following definitions of computational hard problems that form the security foundation of our scheme are presented here.

*Definition 1:* Elliptic curve discrete logarithm problem (ECDLP): On $E$, two random points $\{P, W\} \in \mathbb{G}$ are given. The computation of $a$ is hard such that $W = aP \in \mathbb{G}$ where $a \in \mathbb{Z}_q^*$ is an unknown random number.

*Definition 2:* Elliptic curve computational Diffie-Hellman problem (ECCDHP): On $E$, random points $\{P, W = aP, R = bP\} \in \mathbb{G}$ are given. The computation of $abP \in \mathbb{G}$ is hard, where $(a, b) \in \mathbb{Z}_q^*$ are two unknown random numbers.

## IV. Framework of the Scheme

In this part, the formal definition and security notions for our CLSS-CPPA scheme are briefly introduced.

### A. Generic Model

The generic CLSS-CPPA scheme is composed of seven algorithms. These are: `Setup`, `RegAIDGen`, `PSKGen`, `SPKGen`, `CLSGen`, `CLSVerify` and `BCLSVerify`.

1) `Setup`: It takes a parameter $k$ from the TRA and KGC to provide system parameters *params* and master secret keys $\alpha$ and $\beta$. The TRA and KGC publish *params*. We do not mention *params* in the following algorithms.
2) `RegAIDGen`: The TRA runs this algorithm by taking an original identity $OID_i$ as an input to provide an anonymous-identity $AID_i$. Note this algorithm is run offline.
3) `PSKGen`: The KGC runs this algorithm by taking $AID_i$ as input and $\beta$ to provide a partial private key $psk_i$.
4) `SPKGen`: A sender vehicle executes this algorithm that takes $psk_i$ and a secret value $\mu_i$ as an input to generate a full private key $sk_i$ and a corresponding public key $pk_i$.
5) `CLSGen`: The sender vehicle executes this algorithms that takes a message $m_i \in \{0, 1\}^*$, $AID_i$, and $sk_i$, and gives a signature $\Theta_i$.
6) `CLSVerify`: A receiver vehicle runs this algorithm by taking $m_i$, $\mathcal{A}ID_i$, $\Theta_i$, and $pk_i$, where as $i = 1$ and accepts $m_i$ if $\Theta_i$ is valid; otherwise, it rejects $m_i$.
7) `BCLSVerify`: A receiver vehicle runs this algorithm that takes messages $m_i = \{m_1, m_2, \ldots, m_n\}$, anonymous-identities $AID_i$=$\{AID_1, AID_2, \ldots, AID_n\}$, signatures $\Theta_i$=$\{\Theta_1, \Theta_2, \ldots, \Theta_n\}$, and public keys $pk_i$=$\{pk_1, pk_2, \ldots, pk_n\}$ simultaneously, where

$i = 1, 2, \ldots, n$ and accepts $m_i$ if $\Theta_i$ is valid; otherwise, it rejects $m_i$.

### B. Security Notions

There are two security types in the CLC that belong to type-I attacker and type-Il attacker are considered [33]. These attackers are differentiated with unique capabilities, which are as follows:

• *Type-I Attacker:* It simulates a common user as a malicious user to replace the public key of any user adaptively with a value that it chosen. However, this type of attacker cannot get the KGC master secret key.

• *Type-II Attacker:* This kind of attacker simulates a curious but honest KGC and can approach the KGC master key. But it cannot replace the public key of any user with a value of its own.

We considered that a signature scheme possesses a security notion of EUF-CMA [25]. Therefore, two security notions, i.e., EUF-CMA-I for a type-I attacker and EUF-CMA-II for a type-II attacker for the CLSS-CPPA scheme are assumed. To satisfy these two security notions for our scheme, two games are played, which are illustrated as follows:

*Game-I:* Suppose $\mathcal{F}_a^I$ and $\mathcal{S}$ is a type-I attacker and a simulator, respectively and the game is played between them. In interaction between $\mathcal{F}_a^I$ and $\mathcal{S}$, $\mathcal{S}$ notes all queries with answers in a list.

$Initial$: The `Setup` algorithm takes a security parameter $k$ from simulator $\mathcal{S}$ to generate system parameters *params*. $\mathcal{S}$ provides *params* to $\mathcal{F}_a^I$.

$Attack$: $\mathcal{F}_a^I$ executes the following oracles queries in an adaptive way.

- Partial private key generate queries: $\mathcal{F}_a^I$ submits this query for $AID_i$, $\mathcal{S}$ runs `PPKGen` algorithm to forward a partial private key $spk_i$ to $\mathcal{F}_a^I$.
- Private key generate queries: $\mathcal{F}_a^I$ submits this query for $AID_i$, $\mathcal{S}$ runs `SPKGen` algorithm and provides a private key $sk_i$ to $\mathcal{F}_a^I$.
- Public key request queries: $\mathcal{F}_a^I$ submits this query for $AID_i$, $\mathcal{S}$ runs `SPKGen` algorithm and sends a public key $pk_i$ to $\mathcal{F}_a^I$.
- Public key replace queries: $\mathcal{F}_a^I$ makes this query for $AID_i$ by picking a secret value $\mu_i'$, which it selected already and then sets $pk_i'$ as the new public key. $\mathcal{S}$ records these replacements, which are used later.
- Signing queries: $\mathcal{F}_a^I$ submits this query to sign a message $m_i$ for $AID_i$, $\mathcal{S}$ retrieves $sk_i$ related to $AID_i$, runs `CLS-Gen` algorithm and gives a signature $\Theta_i$ to $\mathcal{F}_a^I$. If $\mathcal{S}$ found that $pk_i$ has been replaced by $\mathcal{F}_a^I$, then $\mathcal{S}$ cannot compute $sk_i$. Therefore, the signing oracle will answer wrong. In that case, we assume that $\mathcal{F}_a^I$ provides an additional secret value $\mu_i'$ to the signing oracle.

$Forgery$: Eventually, $\mathcal{F}_a^I$ gives $\Theta_i^*$ on $m_i^*$ for $AID_i^*$ with a corresponding $pk_i^*$ as output, where $AID_i^*$ is the target challenge identity. $\mathcal{F}_a^I$ can become successful in Game-I if

- $\Theta_i^*$ is a valid signature on $m_i^*$ for $AID_i^*$ and $pk_i^*$.
- $\mathcal{F}_a^I$ has not been asked both partial private key generate queries and private key generate queries for $AID_i^*$.
- $\mathcal{F}_a^I$ has never performed signing queries on $m_i^*$ for $AID_i^*$ and $pk_i^*$.

*Definition 3:* A CLSS-CPPA scheme is EUF-CMA-I secure, if there is any probabilistic polynomial time (PPT) type-I attacker $\mathcal{F}_a^I$, whose success probability $Succ_{\mathcal{F}_a}^k$ for winning the Game-I is negligible.

*Game-II:* Suppose $\mathcal{F}_a^{II}$ and $\mathcal{S}$ is a type-II attacker and a simulator, respectively and the game is played between them. In interaction between $\mathcal{F}_a^{II}$ and $\mathcal{S}$, $\mathcal{S}$ notes all queries with answers in a list.

$Initial$: The `Setup` algorithm takes a security parameter $k$ from $\mathcal{S}$ to generate system parameters *params* and master secret key $\beta$. $\mathcal{S}$ provides both *params* and $\beta$ to $\mathcal{F}_a^{II}$.

$Attack$: $\mathcal{F}_a^{II}$ makes private key generate queries, public key request queries, and signing queries in an adaptive way as performed in Game-I.

$Forgery$: Finally, $\mathcal{F}_a^{II}$ gives $\Theta_i^*$ on $m_i^*$ for $AID_i^*$ with a corresponding $pk_i^*$ as output. $\mathcal{F}_a^{II}$ can become successful in Game-II if

- $\Theta_i^*$ is a valid signature on $m_i^*$ for $AID_i^*$ and $pk_i^*$.
- $\mathcal{F}_a^{II}$ has not been asked private key generate queries for $AID_i^*$.
- $\mathcal{F}_a^{II}$ has never performed signing queries on $m_i^*$ for $AID_i^*$ and $pk_i^*$.

*Definition 4:* A CLSS-CPPA scheme is EUF-CMA-II, if there is any PPT type-II attacker $\mathcal{F}_a^{II}$, whose success probability $Succ_{\mathcal{F}_a}^k$ for winning the Game-II is negligible.

TABLE I
DESCRIPTION OF DIFFERENT NOTATIONS

| Notation | Description |
|---|---|
| KGC | Key Generation Center |
| TRA | Tracing Authority |
| RSU | Road-Side Unit |
| OBU | On-Board Unit |
| $\mathcal{V}_i$ | Sender vehicle |
| $\mathcal{V}_j$ | Receiver vehicle |
| $k$ | Security parameter |
| $\mathbb{G}$ | Additive cyclic group |
| $q$ | Order of $\mathbb{G}$ |
| $P$ | Generator of $\mathbb{G}$ |
| $\{\alpha, T_{pub}\}$ | TRA's master secret and public keys |
| $\{\beta, P_{pub}\}$ | KGC's master secret and public keys |
| $\{OID_i, AID_i\}$ | $\mathcal{V}_i$'s original and anonymous identities |
| $t_i, T_i$ | Valid time periods |
| $psk_i$ | Vehicle $\mathcal{V}_i$'s partial private key |
| $\{sk_i, pk_i\}$ | Vehicle $\mathcal{V}_i$'s private and public keys |
| $m_i$ | Safety message |
| $\{H_0(\cdot), H_1(\cdot), H_2(\cdot)\}$ | General one-way hash functions |
| $\pi$ | Number of bits in an $OID_i$ |
| $\oplus$ | Exclusive-OR operator |
| $\Theta_i$ | Signature |

## V. CLSS-CPPA SCHEME

In this section, the CLSS-CPPA scheme for V2V communication in VANETs is designed. We discuss each algorithm in detail within the context of a VANET. Table I provides the notations used in the CLSS-CPPA scheme.

### *A.* Setup

Both TRA and KGC inputs a security parameter $1^k$ for $k \in N$ to the Setup algorithm to receive the parameters $\{\mathbb{G}, q, P\}$ where $\mathbb{G}$ is the cyclic additive group of prime order $q$ and generator $P$ as described in Section III. Both TRA and KGC performs as follows:

1) The TRA picks a number $\alpha \in \mathbb{Z}_q^*$ randomly as its master secret key and computes a corresponding public key as $T_{pub} = \alpha P$.
2) The KGC also picks a number $\beta \in \mathbb{Z}_q^*$ randomly as its master secret key and computes a corresponding master public key as $P_{pub} = \beta P$.
3) Both TRA and KGC choose $H_0 : \mathbb{G} \to \{0,1\}^\pi$ (where $\pi$ indicates fixed bits' number), $H_1 : \mathbb{G} \times \{0,1\}^\pi \times \mathbb{G} \times \mathbb{G} \to \mathbb{Z}_q^*$, and $H_2 : \{0,1\}^* \times \mathbb{G} \times \{0,1\}^\pi \times \mathbb{G} \times \mathbb{G} \times \mathbb{G} \to \mathbb{Z}_q^*$ as the three random cryptographic one way general hash functions.
4) They then publicly publish the system parameters *params*=$\{q, P, \mathbb{G}, T_{pub}, P_{pub}, H_0, H_1, H_2\}$ across VANET and keep secret $\alpha$ and $\beta$ with themselves. Note that each vehicle's TPD is loaded with the *params* before anonymous-identity and keys generation.

### *B.* RegAIDGen

Through this algorithm, a vehicle $\mathcal{V}_i$ registers itself with the TRA. The vehicle $\mathcal{V}_i$ first submits its original identity $OID_i \in \{0,1\}^\pi$ together with other information, which the vehicle $\mathcal{V}_i$ has obtained from the motor vehicle department (MVD) to the TRA. The vehicle $\mathcal{V}_i$ and TRA accomplish this via the following algorithm:

1) The vehicle $\mathcal{V}_i$ chooses a random number $\gamma_i \in \mathbb{Z}_q^*$.
2) It computes $AID_{i,1} = \gamma_i P$ and forwards $\{OID_i, AID_{i,1}\}$ to the TRA through a secure channel.
3) The TRA verifies the uniqueness of the $OID_i$ from the MVD and computes $AID_{i,2} = OID_i \oplus H_0(\alpha AID_{i,1})$ and sets an anonymous-identity $AID_i = \{AID_{i,1}, AID_{i,2}, T_i\}$, where $T_i$ is the time-stamp that indicates the validity of $AID_i$.
4) It loads the $AID_i$ to the OBU of the vehicle $\mathcal{V}_i$ and sends it to the KGC as well. In addition, the TRA stores it in the database with high security level in order to trace it in the future in case of a dispute. Note the above operations are performed offline.

### *C.* PSKGen

When the vehicle $\mathcal{V}_i$ enters the VANET environment, it sends the anonymous-identity $AID_i$ to the KGC through the RSU for a partial private key generation. The KGC compares $AID_i$ with the one already received from the TRA. If they are equal, then KGC performs as follows:

1) The KGC chooses $\kappa_i \in \mathbb{Z}_q^*$ randomly.
2) Computes $U_i = \kappa_i P$
3) Computes $\theta_i = H_1(AID_i, U_i, P_{pub})$
4) Computes $\lambda_i = (\kappa_i + \theta_i \beta) \bmod q$
5) It then sets $psk_i = \{\lambda_i, U_i\}$ as a partial private key and transmits it to vehicle $\mathcal{V}_i$ through a secure channel.

### *D.* SPKGen

To generate a private key and a corresponding public key, the vehicle $\mathcal{V}_i$ receives a anonymous-identity $AID_i$ and a partial private key $psk_i$. It first computes $\theta_i = H_1(AID_i, U_i, P_{pub})$ and then checks the authenticity of $psk_i$ by verifying the equation $\lambda_i P = U_i + \theta_i P_{pub}$. *Proof of correctness:* The equation $\lambda_i P = U_i + \theta_i P_{pub}$ is verified as follows:

$$\begin{aligned}\lambda_i P &= (\kappa_i + \theta_i \beta) P \\ &= \kappa_i P + \theta_i \beta P \\ &= U_i + \theta_i P_{pub}\end{aligned}$$

The vehicle $\mathcal{V}_i$ does not accept $psk_i$ if $\lambda_i P = U_i + \theta_i P_{pub}$ does not hold; otherwise, it works as follows:

1) The vehicle $\mathcal{V}_i$ chooses a secret value $\mu_i \in \mathbb{Z}_q^*$ randomly and sets its private key $sk_i = \{\mu_i, \lambda_i\}$.
2) Computes $X_i = \mu_i P_{pub}$
3) Computes $Y_i = \lambda_i P_{pub}$
4) The vehicle $\mathcal{V}_i$ sets its public key $pk_i = \{X_i + Y_i\}$ and transmits it to a nearby vehicle.

For our CLSS-CPPA scheme, we suggest a method similar to the one used in [17] to preload the TPD with $AID_i$ and $psk_i$, where $i = 1, 2, \ldots, n$. This loading enables the concerned vehicle $\mathcal{V}_i$ to utilize a unique $AID_i$ and a $psk_i$ each time. After running RegAIDGen and PSKGen algorithms, a large number of $AID_i$ and $psk_i$ with short expiration times are loaded into the vehicle $\mathcal{V}_i$ TPD by the TAs. When all the $AID_i$ and $psk_i$ are used up, the vehicle $\mathcal{V}_i$ then reconnects with the TAs after authentication in its range in order to replenish its stock of $AID_i$ and $psk_i$ through a secure channel.

### *E.* CLSGen

The vehicle $\mathcal{V}_i$ receives a safety message $m_i \in \{0,1\}^*$ from some source (nearby vehicle or sensor) and performs as follows:

1) It chooses a random number $a_i \in \mathbb{Z}_q^*$.
2) Computes $A_i = a_i P_{pub}$
3) Computes $\delta_i = H_2(m_i, AID_i, pk_i, A_i, P_{pub})$
4) It then computes $\eta_i = \delta_i(a_i + \mu_i + \lambda_i) \bmod q$
5) It then sets the signature $\Theta_i = \{\eta_i, A_i\}$ on the message $m_i$ and transmits the message-signature tuple $\{m_i, AID_i, pk_i, \Theta_i, t_i\}$ to a nearby vehicle.

### *F.* CLSVerify

Once a receiver vehicle $\mathcal{V}_j$ receives the tuple $\{m_i, AID_i, pk_i, \Theta_i, t_i\}$ from the vehicle $\mathcal{V}_i$, where $i = 1$, it first ensures the freshness of time-stamps $T_i$ and $t_i$ in the $AID_i$ and $\{m_i, AID_i, pk_i, \Theta_i, t_i\}$, respectively that either these are in the valid time intervals. If these are valid, further verification is continued by the vehicle $\mathcal{V}_j$. The vehicle $\mathcal{V}_j$ performs as follows:

1) It computes $\delta_i = H_2(m_i, AID_i, pk_i, A_i, P_{pub})$

2) The vehicle $\mathcal{V}_j$ accepts $m_i$ if $\Theta_i$ is valid. The validity of $\Theta_i$ can be verified if the equation $\eta_i P_{pub} = \delta_i(A_i + pk_i)$ holds; Otherwise, the vehicle $\mathcal{V}_j$ rejects $m_i$.

*Proof of correctness:* The equation $\eta_i P_{pub} = \delta_i(A_i + pk_i)$ can be verified as follows:

$$\begin{aligned}\eta_i P_{pub} &= \delta_i(a_i + \mu_i + \lambda_i)P_{pub} \\ &= \delta_i\left(a_i P_{pub} + \mu_i P_{pub} + \lambda_i P_{pub}\right) \\ &= \delta_i\left(A_i + X_i + Y_i\right) \\ &= \delta_i(A_i + pk_i)\end{aligned}$$

Therefore, we proved the correctness of a single signature verification.

### G. BCLSVerify

In batch signature verification, the receiver vehicle ensures the authenticity and integrity of multiple safety messages transmitted from a large number of vehicles simultaneously. Therefore, the overall performance of V2V communication is improved. When a vehicle $\mathcal{V}_j$ receives multiple tuples $\{m_1, AID_1, pk_1, \Theta_1, t_1\}$, $\{m_2, AID_2, pk_2, \Theta_2, t_2\}$,...,$\{m_n, AID_n, pk_n, \Theta_n, t_n\}$ sent from vehicles $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_n$, where as $i = 1, 2, \ldots, n$, it first ensures the freshness of time-stamps $T_i$ and $t_i$ in the $AID_i$ and $\{m_i, AID_i, pk_i, \Theta_i, t_i\}$, respectively that either these are in the valid time intervals. If these are valid, further verification is continued by the vehicle $\mathcal{V}_j$. The vehicle $\mathcal{V}_j$ performs as follows:

1) It computes $\delta_i = H_2(m_i, AID_i, pk_i, A_i, P_{pub})$
2) The vehicle $\mathcal{V}_j$ accepts $m_i$ if $\Theta_i$ is valid. The validity of $\Theta_i$ can be verified if the equation $(\sum_{i=1}^{n} \eta_i)P_{pub} = \sum_{i=1}^{n} \delta_i(A_i + pk_i)$ holds; Otherwise, the vehicle $\mathcal{V}_j$ rejects $m_i$.

*Proof of correctness:* The equation $(\sum_{i=1}^{n} \eta_i)P_{pub} = \sum_{i=1}^{n} \delta_i(A_i + pk_i)$ can be verified as follows:

$$\begin{aligned}\left(\sum_{i=1}^{n} \eta_i\right) P_{pub} &= \sum_{i=1}^{n} \delta_i\left(a_i + \mu_i + \lambda_i\right) P_{pub} \\ &= \sum_{i=1}^{n} \delta_i\left(a_i P_{pub} + \mu_i P_{pub} + \lambda_i P_{pub}\right) \\ &= \sum_{i=1}^{n} \delta_i\left(A_i + X_i + Y_i\right) \\ &= \sum_{i=1}^{n} \delta_i(A_i + pk_i)\end{aligned}$$

Therefore, we proved the correctness of batch signature verification.

## VI. Security Analysis

We provide security proof and also discuss the necessary security requirements of the proposed CLS-CPPA scheme in this section.

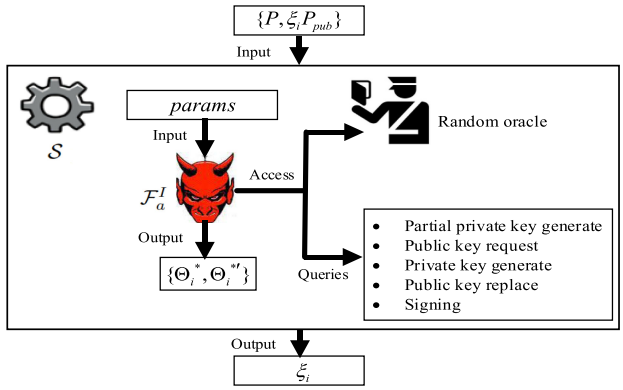


Fig. 2. Structure for the security proof of Lemma 1.

### A. Security Proof

Based on the following Theorem, the security proof of the CLSS-CPPA scheme is obtained.

*Theorem 1:* The CLSS-CPPA scheme is EUF-CMA secure against type-I attacker and type-II attacker in the ROM with a notion that the ECDLP is unbreakable.

According to Definitions 3 and 4, Theorem 1 is proved with help of Lemmas 1 and 2.

*Lemma 1:* If there is a PPT type-I attacker $\mathcal{F}_a^I$ having a non-negligible advantage $\phi$ against CLSS-CPPA scheme with respect to EUF-CMA-I as discussed in Game-1 after executing $q_{H_1}$ queries, $q_{H_2}$ queries, $q_{psk}$ partial private key generate queries, $q_{sk}$ private key generate queries, $q_{pk}$ public key request queries, and $q_{sig}$ signing queries in a time $t$, then there exists a simulator $\mathcal{S}$ that can solve the ECDLP in time $t'$ expected to be less than $120686 q_{H_1} q_{H_2} t'/\phi$, if $\phi \geq 10(q_{sig} + 1)(q_{H_1} + q_{H_2} + q_{psk} + q_{sk} + q_{pk} + q_{sig})/q$.

*Proof:* Suppose a type-I attacker $\mathcal{F}_a^I$, who acts as a forger against the proposed CLSS-CPPA scheme. We show how a simulator $\mathcal{S}$ exploits the ability of $\mathcal{F}_a^I$ to solve the ECDLP. $\mathcal{S}$ plays the role of challenger and accepts a challenge containing a random instance of $\{P, W\} \in \mathbb{G}$, where $W = \xi_i P_{pub}$ and $\xi_i \in \mathbb{Z}_q^*$ as shown in Fig. 2. $\mathcal{S}$ task is to compute $\xi_i$.

In this simulation, we consider the hash functions $H_1$ and $H_2$ as random oracles, which are queried by $\mathcal{F}_a^I$. We know that $\mathcal{F}_a^I$ performs $H_1$ query for any anonymous-identity $AID_i$ before performing other queries. Two hash lists $L_{H_1}$ and $L_{H_2}$ and one public key list $L_{pk}$ are maintained by $\mathcal{S}$ to keep record of all queries and answer in interaction with $\mathcal{F}_a^I$. Lists $L_{H_1}$, $L_{H_2}$ and $L_{pk}$ are considered empty in the initial stage. $\mathcal{S}$ answers all the queries that are asked by $\mathcal{F}_a^I$ in this simulation, which are as follows:

- Setup: To generate public parameters *params*=$\{q, P, \mathbb{G}, P_{pub}, H_1, H_2\}$ and the KGC's master secret key $\beta$, $\mathcal{S}$ executes the Setup. Note $\beta$ is unknown to $\mathcal{S}$. It sets $AID_i^*$ randomly as a challenge anonymous-identity for $\mathcal{F}_a^I$ in this game and sends *params* to $\mathcal{F}_a^I$.
- $H_1$ queries: $\mathcal{F}_a^I$ submits this query with an input $\{AID_i, U_i\}$, $\mathcal{S}$ looks up the list $L_{H_1}$ for the input. If this query for $AID_i$ is already defined in the list $L_{H_1}$, $\mathcal{S}$ returns the hash value $H_1$ to $\mathcal{F}_a^I$; otherwise, $\mathcal{S}$ simulates the

oracle as follows: $\mathcal{S}_I$ selects $\kappa_i \in \mathbb{Z}_q^*$ randomly, computes $U_i = \kappa_i P$ and $\theta_i = H_1(AID_i, U_i, P_{pub})$. $\mathcal{S}$ forwards $\theta_i$ to $\mathcal{F}_a^I$ and adds the elements $\{AID_i, U_i, \theta_i\}$ to the list $L_{H_1}$.

- Partial private key generate queries: $\mathcal{F}_a^I$ makes this query for $AID_i$, $\mathcal{S}$ performs as follows: If $AID_i = AID_i^*$, $\mathcal{S}$ then aborts the simulation. If $AID_i \neq AID_i^*$, $\mathcal{S}$ chooses two numbers $\{\vartheta_i, \psi_i\} \in \mathbb{Z}_q^*$ randomly, computes $U_i = \vartheta_i P - \psi_i P$. It sets $\lambda_i = \vartheta_i$ and $H_1(PID_i, U_i, P_{pub}) = \theta_i = \psi_i$. $\mathcal{S}$ sets a partial private key as $psk_i = \{\lambda_i, U_i\}$. $\mathcal{S}$ then transmits $psk_i$ to $\mathcal{F}_a^I$ and adds the elements $\{AID_i, U_i, \theta_i, \lambda_i\}$ to the list $L_{H_1}$.
- Public key request queries: $\mathcal{F}_a^I$ submits this query for $AID_i$. $\mathcal{S}$ looks up the list $L_{pk}$ to see whether this query is already available for the $AID_i$. If it is, then $\mathcal{S}$ provides the public key $pk_i$ to $\mathcal{F}_a^I$. Otherwise, $\mathcal{S}$ recovers the corresponding elements $\{AID_i, U_i, \theta_i, \lambda_i\}$ from the list $L_{H_1}$, chooses an integer $\xi_i \in \mathbb{Z}_q^*$ randomly, computes $X_i = \xi_i P_{pub}$ and $Y_i = \lambda_i P_{pub}$. $\mathcal{S}$ sends the public key $pk_i = \{X_i + Y_i\}$ to $\mathcal{F}_a^I$ and adds the elements $\{AID_i, \lambda_i, \xi_i, pk_i\}$ to the list $L_{pk}$.
- Private key generate queries: $\mathcal{F}_a^I$ submits this query for $AID_i$, $\mathcal{S}$ first checks whether $AID_i = AID_i^*$. If $AID_i = AID_i^*$, then $\mathcal{S}$ aborts the simulation; otherwise, $\mathcal{S}_I$ performs as follows: $\mathcal{S}$ looks up the list $L_{pk}$ to see that whether this query is already available for the $AID_i$. If it is, then $\mathcal{S}$ sends that private key $sk_i$ to $\mathcal{F}_a^I$. Otherwise, $\mathcal{S}$ uses the partial private key generate and the public key request queries to generate $\{AID_i, \lambda_i, \xi_i, X_i, Y_i\}$. $\mathcal{S}$ then performs as the above process and sends the private key $sk_i = \{\xi_i, \lambda_i\}$ to $\mathcal{F}_a^I$.
- Public key replace queries: $\mathcal{F}_a^I$ submits this query with an input $\{AID_i, pk_i'\}$, where $pk_i' = \{X_i' + Y_i'\}$, $X_i' = \mu_i' P_{pub}$ and $Y_i' = \lambda_i' P_{pub}$. $\mathcal{S}$ sets $X_i = X_i'$, $Y_i = Y_i'$, $\lambda_i = \lambda_i'$ and $\xi_i = \xi_i'$ and adds $\{AID_i, \lambda_i', \xi_i', pk_i'\}$ to the list $L_{pk}$.
- $H_2$ queries: $\mathcal{F}_a^I$ submits this query with an input $\{m_i, AID_i, pk_i, A_i\}$, $\mathcal{S}$ looks up the list $L_{H_2}$ that whether this query has already available for $AID_i$. If it does, $\mathcal{S}$ returns the hash value $H_2$ to $\mathcal{F}_a^I$. Otherwise, a random number hash value $H_2(m_i, AID_i, pk_i, A_i, P_{pub}) = \delta_i$ is computed by $\mathcal{S}$. $\mathcal{S}$ then forwards $\delta_i$ to $\mathcal{F}_a^I$ and adds $\{m_i, AID_i, pk_i, A_i, \delta_i\}$ to the list $L_{H_2}$.
- Signing queries: $\mathcal{F}_a^I$ submits this query to sign a message $m_i$ for $AID_i$, two numbers $\{\lambda_i, \delta_i\} \in \mathbb{Z}_q^*$ are selected randomly by $\mathcal{S}$ from the lists $L_{pk}$ and $L_{H_2}$, respectively. Next, it chooses an integer $\zeta_i \in \mathbb{Z}_q^*$ randomly to compute $A_i = \zeta_i P_{pub}$ and set $A_i = \lambda_i P_{pub} - X_i - Y_i$ and $\eta_i = \delta_i \lambda_i \mod q$. It then sets $\Theta_i = \{\eta_i, A_i\}$, sends it to $\mathcal{F}_a^I$ as the response of the signing query and adds $\{m_i, AID_i, pk_i, A_i, \delta_i\}$ to the list $L_{H_2}$. The response to the signing queries satisfies equation $\eta_i P_{pub} = \delta_i(A_i + pk_i)$.

$$\begin{aligned}\eta_i P_{pub} &= \delta_i \lambda_i \left(\frac{A_i + X_i + Y_i}{\lambda_i}\right) \\ &= \delta_i (A_i + X_i + Y_i) \\ &= \delta_i (A_i + pk_i)\end{aligned}$$

Finally, $\mathcal{F}_a^I$ aborts and outputs a signature $\Theta_i^* = \{\eta_i^*, A_i\}$ on a message $m_i^*$ for the target identity $AID_i^*$, which can satisfy the equation below

$$\eta_i^* P_{pub} = \delta_i^* (A_i + pk_i) \tag{1}$$

If $AID_i \neq AID_i^*$, $\mathcal{S}$ outputs "failure" and terminates; otherwise, $\mathcal{S}$ retrieves the elements $\{AID_i, U_i, \theta_i, \lambda_i\}$, $\{AID_i, \lambda_i, pk_i\}$ and $\{m_i, AID_i, pk_i, A_i, \delta_i\}$ from the lists $L_{H_1}$, $L_{pk}$ and $L_{H_2}$, respectively.

By using the Lemma in [43], $\mathcal{F}_a^I$ can provide another valid signature $\Theta^{*\prime} = \{\eta_i^{*\prime}, A_i\}$ if the same process with a different choice of $H_2$ value (i.e., $\delta_i^* \neq \delta_i^{*\prime}$) is repeated. It can also satisfy the following equation.

$$\eta_i^{*\prime} P_{pub} = \delta_i^{*\prime} (A_i + pk_i) \tag{2}$$

According to the two linear independent equations Eqs. 1 and 2, we subtract 2 from 1, we get.

$$\begin{aligned}&\eta_i^* P_{pub} - \eta_i^{*\prime} P_{pub} = \delta_i^* (A_i + pk_i) - \delta_i^{*\prime} (A_i + pk_i) \\ &\eta_i^* P_{pub} - \eta_i^{*\prime} P_{pub} = \delta_i^* (A_i + Y_i + X_i) - \delta_i^{*\prime} (A_i + Y_i + X_i) \\ &(\eta_i^* - \eta_i^{*\prime}) P_{pub} = (\delta_i^* - \delta_i^{*\prime})(A_i + Y_i + X_i) \\ &(\eta_i^* - \eta_i^{*\prime}) P_{pub} = (\delta_i' - \delta_i^{*\prime})(\zeta_i P_{pub} + \lambda_i P_{pub} + \xi_i P_{pub}) \\ &\left(\frac{\eta_i^* - \eta_i^{*\prime}}{\delta_i^* - \delta_i^{*\prime}}\right) P_{pub} = (\zeta_i + \lambda_i + \xi_i) P_{pub} \\ &\frac{\eta_i^* - \eta_i^{*\prime}}{\delta_i^* - \delta_i^{*\prime}} = (\zeta_i + \lambda_i + \xi_i) \bmod q \\ &\left(\frac{\eta_i^* - \eta_i^{*\prime}}{\delta_i^* - \delta_i^{*\prime}}\right) - (\zeta_i + \lambda_i) = \xi_i \bmod q\end{aligned}$$

$\mathcal{S}$ outputs $(\frac{\eta_i^* - \eta_i^{*\prime}}{\delta_i^* - \delta_i^{*\prime}}) - (\zeta_i + \lambda_i)$ as the solution of the ECDLP. Using the above, we draw this conclusion that $\mathcal{S}$ can break the ECDLP. Therefore, with the notion that the ECDLP in $\mathbb{G}$ is unbreakable, the proposed CLSS-CPPA scheme for V2V communication is EUF-CMA-I secure against type-I attacker $\mathcal{F}_a^I$ in the ROM. ■

*Lemma 2:* If there is a PPT type-II attacker $\mathcal{F}_a^{II}$ having a non-negligible advantage $\phi$ against CLSS-CPPA scheme with respect to EUF-CMA-II as discussed in Game-2 after executing $q_{H_1}$ queries, $q_{H_2}$ queries, $q_{sk}$ private key generate queries, $q_{pk}$ public key request queries, and $q_{sig}$ signing queries in time $t$, then there exists a simulator $\mathcal{S}$ that can solve the ECDL problem in time $t'$ expected to be less than $120686 q_{H_1} q_{H_2} t'/\phi$, if $\phi \geq 10(q_{sig}+1)(q_{H_1} + q_{H_2} + q_{sk} + q_{pk} + q_{sig})/q$.

*Proof:* Suppose a type-II attacker $\mathcal{F}_a^{II}$, who acts as a forger against the proposed CLSS-CPPA scheme. We show how a simulator $\mathcal{S}$ exploits the ability of $\mathcal{F}_a^{II}$ to solve the ECDLP. Suppose $\mathcal{S}$ performs the role of challenger and accepts a challenge containing a random instance of $\{P, W\} \in \mathbb{G}$, where $W = P_{pub} = \beta P$ and $\beta \in \mathbb{Z}_q^*$ as shown in Fig 3. The task of $\mathcal{S}$ is to compute $\beta$.

In this simulation, we consider the hash functions $H_1$ and $H_2$ as random oracles, which are queried by $\mathcal{F}_a^{II}$. We know that $\mathcal{F}_a^{II}$ asks $H_1$ query for any anonymous-identity $AID_i$ before performing other queries. $\mathcal{S}$ maintains two hash lists $L_{H_1}$ and

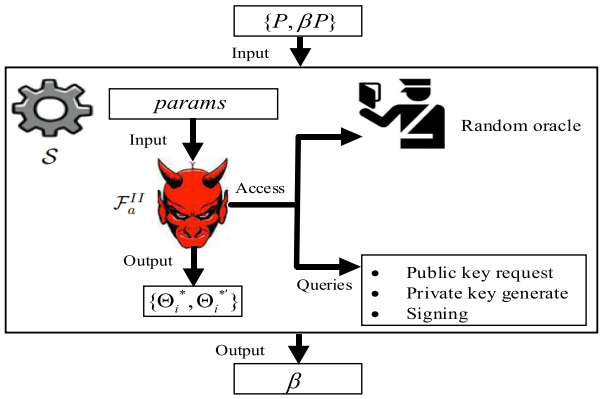


Fig. 3. Structure for the security proof of Lemma 2.

$L_{H_2}$ and one public key list $L_{pk}$. Both $\mathcal{F}_a^{II}$ and $\mathcal{S}$ perform like Game-II in Definition 4, which are as follows:

- Setup: The system parameters $params$=$\{q, P, \mathbb{G}, P_{pub}, H_1, H_2\}$ and the KGC's master secret key $\beta$ are generated as $\mathcal{S}$ executes the Setup. Note $\beta$ is unknown to $\mathcal{S}$. It sets $AID_i^*$ randomly as a challenge anonymous-identity for $\mathcal{F}_a^{II}$ in this game. $\mathcal{S}$ sends $params$ and $\beta$ to $\mathcal{F}_a^{II}$.
- $H_1$ queries: This query is same to the $H_1$ query performed in Lemma 1.
- Public key request queries: $\mathcal{F}_a^{II}$ makes this query for $AID_i$. $\mathcal{S}$ looks up the list $L_{pk}$ to determine whether this query for the $AID_i$ has already been performed. If it has, $\mathcal{S}$ then provides the public key $pk_i$ to $\mathcal{F}_a^{II}$. Otherwise, $\mathcal{S}$ recovers $\{AID_i, U_i, \theta_i, \lambda_i\}$ from the list $L_{H_1}$, selects a random value $\mu_i \in \mathbb{Z}_q^*$, computes $X_i = \mu_i P_{pub}$ and $Y_i = \lambda_i P_{pub}$. It then sends the public key $pk_i = \{X_i + Y_i\}$ to $\mathcal{F}_a^{II}$ and adds $\{AID_i, \lambda_i, \mu_i, pk_i\}$ to the list $L_{pk}$.
- Private key generate queries: $\mathcal{F}_a^{II}$ submits this query for $AID_i$. $\mathcal{S}$ first checks whether $AID_i = AID_i^*$. If $AID_i = AID_i^*$, then $\mathcal{S}$ aborts the simulation; otherwise, $\mathcal{S}$ performs as follows: $\mathcal{S}$ looks up the list $L_{pk}$ to determine whether this query has already been performed for the $AID_i$. If it has, $\mathcal{S}$ then provides the private key $sk_i$ to $\mathcal{F}_a^{II}$. Otherwise, $\mathcal{S}$ makes partial private key generate and public key request queries to generate $\{AID_i, \lambda_i, \mu_i, pk_i\}$. It then performs as the above process and transmits $sk_i = \{\mu_i, \lambda_i\}$ as the private key to $\mathcal{F}_a^{II}$.
- $H_2$ queries: This query is same to the $H_2$ query performed in Lemma 1.
- Signing queries: $\mathcal{F}_a^{II}$ submits this query to sign a message $m_i$ for $AID_i$. From lists $L_{pk}$ and $L_{H_2}$, $\mathcal{S}$ picks two random integers $\{\lambda_i, \delta_i\} \in \mathbb{Z}_q^*$, respectively. Next, it chooses an integer $\zeta_i \in \mathbb{Z}_q^*$ randomly to compute $A_i = \zeta_i P_{pub}$ and set $A_i = \lambda_i P_{pub} - X_i - Y_i$ and $\eta_i = \delta_i \lambda_i \mod q$. It then sets $\Theta_i = \{\eta_i, A_i\}$ and sends it to $\mathcal{F}_a^{II}$ as the response of the signing query and adds $\{m_i, AID_i, pk_i, A_i, \delta_i\}$ to the list $L_{H_2}$. The response to the signing queries satisfies equation $\eta_i P_{pub} = \delta_i(A_i + pk_i)$.

Finally, $\mathcal{F}_a^{II}$ aborts and provides a signature $\Theta_i^* = \{\eta_i^*, A_i\}$ on $m_i^*$ for the target identity $AID_i^*$, which can satisfy equation below

$$\eta_i^* P_{pub} = \delta_i^*(A_i + pk_i) \tag{3}$$

If $AID_i \neq AID_i^*$, $\mathcal{S}$ outputs "failure" and terminates; otherwise, $\mathcal{S}$ retrieves $\{AID_i, U_i, \theta_i, \lambda_i\}$, $\{AID_i, \lambda_i, pk_i\}$ and $\{m_i, AID_i, pk_i, \delta_i\}$ from the lists $L_{H_1}$, $L_{pk}$, and $L_{H_2}$, respectively.

By using the Lemma [43], $\mathcal{F}_a^{II}$ can provide another valid signature $\Theta_i^{*\prime} = \{\eta_i^{*\prime}, A_i\}$ if the same process with a different choice of $H_2$ (i.e., $\delta_i^* \neq \delta_i^{*\prime}$) is repeated. It can also satisfy the following equation.

$$\eta_i^{*\prime} P_{pub} = \delta_i^{*\prime}(A_i + pk_i) \tag{4}$$

According to the two linear independent Eqs. 3 and 4, we subtract 4 from 3, we get.

$$\eta_i^* P_{pub} - \eta_i^{*\prime} P_{pub} = \delta_i^*(A_i + pk_i) - \delta_i^{*\prime}(A_i + pk_i)$$

$$\eta_i^* P_{pub} - \eta_i^{*\prime} P_{pub} = \delta_i^*(A_i + Y_i + X_i) - \delta_i^{*\prime}(A_i + Y_i + X_i)$$

$$(\eta_i^* - \eta_i^{*\prime}) P_{pub} = (\delta_i^* - \delta_i^{*\prime})(A_i + Y_i + X_i)$$

$$\left(\frac{\eta_i^* - \eta_i^{*\prime}}{\delta_i^* - \delta_i^{*\prime}}\right) P_{pub} = \zeta_i P_{pub} + \lambda_i P_{pub} + \mu_i P_{pub}$$

$$\left(\frac{\eta_i^* - \eta_i^{*\prime}}{\delta_i^* - \delta_i^{*\prime}}\right) \beta P = (\zeta_i + \lambda_i + \mu_i)\beta P$$

$$\frac{(\eta_i^* - \eta_i^{*\prime})\beta}{(\delta_i^* - \delta_i^{*\prime})(\zeta_i + \lambda_i + \mu_i)} = \beta \mod q$$

$\mathcal{S}$ outputs $\frac{(\eta_i^* - \eta_i^{*\prime})\beta}{(\delta_i^* - \delta_i^{*\prime})(\zeta_i + \lambda_i + \mu_i)}$ as the solution to the ECDLP. We conclude that $\mathcal{S}$ can break the ECDLP. Therefore, the CLSS-CPPA scheme is secure with respect to EUF-CMA against type-II attacker $\mathcal{F}_a^{II}$ with the notion that ECDLP in $\mathbb{G}$ is unbreakable in the ROM. ■

### B. Security Requirements

Our CLSS-CPPA scheme ensures the following security requirements in VANETs.

1) *Message authentication and integrity:* In the CLSS-CPPA scheme, message $m_i$ is authenticated and its integrity is checked by verifying $\eta_i P_{pub} = \delta_i(A_i + pk_i)$. If it holds, $m_i$ is accepted; otherwise, $m_i$ is rejected. In addition, in Section VI-A, we proved that our CLSS-CPPA scheme provides security against type-I and type-II attackers with respect to EUF-CMA as well as adaptive chosen identity attack with the notion that ECDLP (in Definition 1) is hard in the ROM. Therefore, message is authenticated and integrity is checked by our scheme.
2) *Privacy (identity-anonymity):* The anonymous-identity $AID_i$ in our scheme, consists of the secret key chosen by the vehicle $\mathcal{V}_i$ and TRA chosen secret key $\{\gamma_i, \alpha\}$, respectively are only known to vehicle $\mathcal{V}_i$ and the TRA. According to Definition 2, no PPT attacker is able to compute $\alpha AID_{i,1}$ in order to obtain the original identity $OID_i$ of the vehicle $V_i$ without prior knowledge of the values of $\alpha$ and $\gamma_i$. To find $OID_i$ from $AID_i = \{AID_{i,1}, AID_{i,2}\}$,

where $AID_{i,2} = OID_i \oplus H_0(\alpha AID_{i,1}, T_{pub})$, the attacker has to compute $\alpha AID_{i,1} = \alpha_i \gamma_i P$ from $T_{pub} = \alpha P$ and $AID_{i,1} = \gamma_i P$. It means that no attacker can extract $OID_i$ from the $AID_i$ due to the ECCDHP in Definition 2. Therefore, identity privacy is preserved by our scheme.

3) *Non-repudiation:* A vehicle $V_i$ in the CLSS-CPPA scheme cannot deny a message which it has generated. Because the vehicle $V_i$ is registered with the TRA database. If the vehicle $V_i$ tries to deny any message which it had sent. The TRA will identify it through its $AID_i$. $AID_i$ shows the source of the generated message. Therefore, non-repudiation in V2V communication is ensured by CLSS-CPPA scheme.
4) *Conditional traceability:* In the CLSS-CPPA scheme, in case of disputed message (i.e., fake signature/message is detected), only the TRA can obtain the original identity $OID_i$ of a vehicle $\mathcal{V}_i$ by using his master secret key $\alpha$ as follows:

$$\begin{aligned} OID_i &= AID_{i,2} \oplus H_0(\alpha AID_{i,1}, T_{pub}) \\ &= OID_i \oplus H_0(\alpha AID_{i,1}, T_{pub}) \oplus H_0(\alpha AID_{i,1}, T_{pub}) \\ &= OID_i \end{aligned}$$

Hence, $OID_i$ of the vehicle $\mathcal{V}_i$ is recovered. Therefore, traceability is ensured by our scheme.
5) *Unlinkability:* In our scheme, a malicious vehicle cannot link message $m$ and message $m'$ whether these are generated from the same vehicle or not. This is because it cannot compute the ECCDHP in Definition 2. Different private keys $sk_i$, where $i = 1, 2, \ldots, n$ are used for signing these messages. The partial private keys $psk_i$ are based on anonymous identities $AID_i$ and there does not exist any link between them. The private keys $sk_i$ and also the signature $\Theta_i$ are constructed from secure random numbers. For instance, a vehicle $\mathcal{V}_i$ computes $AID_{i,1}{=}\gamma_i P$, where $\gamma_i \in \mathbb{Z}_q^*$, the KGC computes $U_i{=}\kappa_i P$ and $\lambda_i{=}\kappa_i + \theta_i \beta \bmod q$ where $\{\kappa_i, \beta\} \in \mathbb{Z}_q^*$, and the vehicle $\mathcal{V}_i$ computes its private key $sk_i{=}\{psk_i, \mu_i\}$, where $\mu_i \in \mathbb{Z}_q^*$. Due to these random numbers, no malicious vehicle can be able to link message $m$ and message $m'$ and know that they were generated from the same vehicle.
6) *Partial distribution of authority:* In the CLC, the authority of a private key generation by the KGC is partially distributed to the user. In the proposed CLSS-CPPA scheme, the KGC generates the partial private key $psk_i$. The vehicle $\mathcal{V}_i$ computes a full private key $sk_i$ by using $psk_i$ and a random secret value $\mu_i \in \mathbb{Z}_q^*$ chosen by himself. Thus, the KGC cannot sign any message on behalf of the vehicle $\mathcal{V}_i$ because it does not have the secret value $\mu_i$. Therefore, the inherent key escrow problem is solved.
7) *Resistance against attacks:* The CLSS-CPPA scheme protects V2V communication from the following attacks:
*Impersonation attacks:* In our scheme, no one can compose a message-signature tuple $\{m_i, AID_i, \Theta_i, t_i\}$ on behalf of vehicle $\mathcal{V}_i$ to vehicle $\mathcal{V}_j$ according to Theorem 1. This is because the vehicle $\mathcal{V}_j$ authenticates the tuple $\{m_i, AID_i, \Theta_i, t_i\}$ and can detect impersonation attacks easily by checking $\eta_i P_{pub} = \delta_i(A_i + pk_i)$. If holds, it accepts the message $m_i$; otherwise, $m_i$ is rejected. Hence, our CLSS-CPPA scheme resists impersonation attacks.
*Modification attacks:* In our scheme, any modification in the message-signature tuple $\{m, AID_i, \Theta_i, t_i\}$ can be identified by verifying $\eta_i P_{pub} = \delta_i(A_i + pk_i)$. If holds, the receiver vehicle $\mathcal{V}_j$ accepts the message $m_i$; otherwise, $m_i$ is rejected. Therefore, our scheme can resist modification attacks.
*Man-in-the-middle attacks:* The CLSS-CPPA scheme ensures authentication efficiently and that no any third party is involved between a message signing and verification at both sides. Therefore, message authentication is only performed between sender and receiver. Hence, there can be no risk of man-in-the-middle attacks.
*Replay Attacks:* The time-stamps $T_i$ and $t_i$ are added to $AID_i = \{AID_{i,1}, AID_{i,2}, T_i\}$ and $\{m_i, AID_i, \Theta_i, t_i\}$, respectively to enable vehicle $\mathcal{V}_j$ to detect replay attacks by checking the newness of $T_i$ and $t_i$. Therefore, our CLSS-CPPA scheme resists replay attacks.

## VII. Performance Evaluation

We evaluate the performance of our CLSS-CPPA scheme with respect to two parameters (i.e., computational cost and communication cost) in V2V communication.

### A. Analysis of Computational Cost

We analyze our CLSS-CPPA scheme and the recent related schemes [9], [10], [38]–[41] with respect to computational cost incurred from message signing and verification. For our scheme and the scheme in [9] are based on the ECC, a finite field $\mathbb{F}_p$ as mentioned in Section III is considered. We set 80 bits security level, then $p$ and $q$ are two 160 bits prime numbers. For the bilinear pairings-based schemes in [10], [38]–[41], we utilize a bilinear pairing $\hat{e}$ such that $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ and set 80 bits security level. Here $\mathbb{G}_1$ is the additive group with a large prime order $\overline{q}$ generated by a point $\overline{P}$ on elliptic curve $\overline{E}$. It uses an equation $y^2 \equiv (x^3 + x) \mod \overline{p}$ having 2 is the embedded degree, then $\overline{p} = 512$ bits and $\overline{q} = 160$ bits prime numbers.

In this analysis, we address those operations that take significant amount of time to process in the whole algorithm. Let the operations $P_{bp}$, $M_{bp}$, $A_{bp}$, $H_{mtp}$, $M_{ecc}$, and $A_{ecc}$ denote time required for a bilinear pairing, a scalar multiplication in $\mathbb{G}_1$, a point addition in $\mathbb{G}_1$, a map-to-point hash function in $\mathbb{G}_1$, a scalar multiplication in $\mathbb{G}$, and a point addition in $\mathbb{G}$, respectively. Note, the execution time required for a general hash function operation, XOR operation, and general multiplication are not considered because these take negligible amount of time to process. We adopt the experiment and the method of evaluation in [25]. According to the experiment in [25], we show the execution time of each of the aforementioned operations in ms in Table II.

According to Table III, a vehicle $\mathcal{V}_i$ through Tsai's scheme [40] requires one operation of scalar multiplication in $G_1$ to sign a message $m_i$ while a receiver vehicle $\mathcal{V}_j$ needs operations of one bilinear pairing, two scalar multiplications

TABLE II
EXECUTION TIMES OF CRYPTOGRAPHIC OPERATIONS

| Cryptographic Operation: | $P_{bp}$ | $M_{bp}$ | $A_{bp}$ | $H_{mtp}$ | $M_{ecc}$ | $A_{ecc}$ |
|---|---|---|---|---|---|---|
| Execution Time (ms): | 4.211 | 1.709 | 0.0071 | 4.406 | 0.442 | 0.0018 |

in $\mathbb{G}_1$, and two point additions in $\mathbb{G}_1$ to verify the concerned signature $\Theta_i$. A vehicle $\mathcal{V}_i$ in Tsai's scheme [40] requires a cost of $1M_{bp} \approx 1.709$ ms in signing the message $m_i$ while to verify the concerned signature $\Theta_i$, the vehicle $\mathcal{V}_j$ needs a cost of $1P_{bp} + 2M_{bp} + 2A_{bp} \approx 7.6432$ ms. Therefore, in both signature $\Theta_i$ generation and verification, a total computational cost incurred by Tsai's scheme [40] is approximately equal to 9.3522 ms. Similarly the vehicle $\mathcal{V}_j$ in Tsai's scheme [40] needs $1P_{bp} + 2nM_{bp} + 2nA_{bp} \approx 4.211 + 3.4322n$ ms cost to verify $n$ signatures $\Theta_i$ where $i = 1, 2, \ldots, n$. In Ali *et al.*'s scheme [10], a vehicle $\mathcal{V}_i$ generates a signature $\Theta_i$ at cost of $1M_{bp} \approx 1.709$ ms while a vehicle $\mathcal{V}_j$ verifies it with a cost of $1P_{bp} + 1M_{bp} + 1A_{bp} \approx 5.9271$ ms. Thus, both message signing and verification in Ali *et al.*'s scheme [10] requires a total computational cost, which is approximately equal to 7.6361 ms. For $n$ signatures $\Theta_i$ verification, the vehicle $\mathcal{V}_j$ in Ali *et al.*'s scheme [10] needs a cost of $1P_{bp} + nM_{bp} + nA_{bp} \approx 4.211 + 1.7161n$ ms. The computational costs of the other bilinear pairing-based schemes in [38], [39], [41] can be calculated in the same way and are shown in Table III.

From the Table III, a message $m_i$ is signed by a vehicle $\mathcal{V}_i$ in Cui *et al.*'s scheme [9], which comprises one operation of scalar multiplication in $\mathbb{G}$ while for the verification of the corresponding signature $\Theta_i$, a receiver vehicle $\mathcal{V}_j$ needs operations of three scalar multiplications in $\mathbb{G}$ and two point additions in $\mathbb{G}$. The vehicle $\mathcal{V}_i$ in Cui *et al.*'s scheme [9] requires a cost of $1M_{ecc} \approx 0.442$ ms in message $m_i$ signing while for verification of the signature $\Theta_i$, the vehicle $\mathcal{V}_j$ needs $3M_{ecc} + 2A_{ecc} \approx 1.3296$ ms cost. Thus, Cui *et al.*'s scheme [9] takes in both message signing and verification, a total computational cost approximately equal to 1.7716 ms. Similarly the vehicle $\mathcal{V}_j$ in Cui *et al.*'s scheme [9] needs $(n+2)M_{ecc} + 2nA_{ecc} \approx 0.844 + 0.4456n$ ms cost to verify $n$ signatures $\Theta_i$. A vehicle $\mathcal{V}_i$ in our CLSS-CPPA scheme, signs a message $m_i$ with one scaler multiplication in $\mathbb{G}$ operation while a receiver vehicle $\mathcal{V}_j$ needs two scalar multiplication operations in $\mathbb{G}$ and one point addition operation in $\mathbb{G}$ to verify the concerned signature $\Theta_i$. Therefore, the vehicle $\mathcal{V}_i$ in our scheme needs a cost of $1M_{ecc} \approx 0.442$ ms in message $m_i$ signing. The corresponding signature $\Theta_i$ is verified with a cost of $2M_{ecc} + 1A_{ecc} \approx 0.8858$ ms by the vehicle $\mathcal{V}_j$. Thus, both message signing and verification in our scheme requires a total computational cost, which is approximately equal to 1.3278 ms. Similarly the vehicle $\mathcal{V}_j$ in our scheme requires a cost of $(n+1)M_{ecc} + nA_{ecc} \approx 0.442 + 0.4438n$ ms to verify $n$ signatures $\Theta_i$. The computational cost of the CLSS-CPPA scheme and the related schemes in [9], [10], [38]–[41] with respect to a message signing, single signature verification, and multiple signatures verification are compared graphically in Figs. 4, 5, and 6, respectively.

Now, the cost of the CLSS-CPPA scheme is compared with the cost of the schemes in [9], [10], [38]–[41] in terms of a message signing, single signature verification, and multiple signatures verification. The percentage improvement of our scheme with respect to Tsai's scheme [40] is about $\frac{1.709-0.442}{1.709} \times 100 \approx 74.14\%$ $\frac{7.6432-0.8858}{7.6432} \times 100 \approx 88.41\%$, and $\frac{4.211+3.4322n-(0.442+0.4438n)}{4.211+3.4322n} \times 100 \approx 87.09\%$, respectively, where $n$ denotes the total number of signatures, which is 120. For the other schemes in [9], [10], [38], [39], [41], the improvement with respect to percentage are given in Table IV. From Table IV and Figs. 4, 5, and 6, it is observed that our scheme indicates high efficiency in terms of a message signing, single signature verification, and multiple signatures verification in comparison with the schemes in [10], [38]–[41]. However, our scheme does not show much improvement in

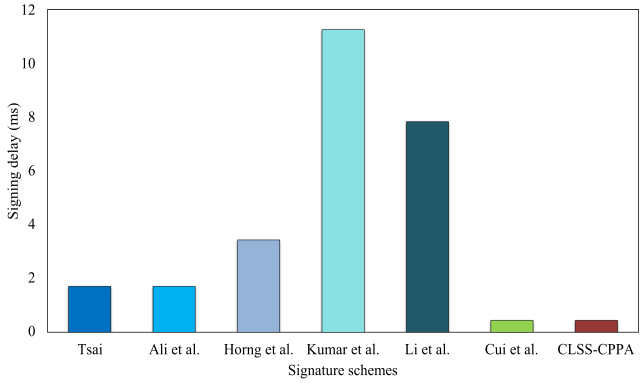


Fig. 4. Computational cost of a single message signing.

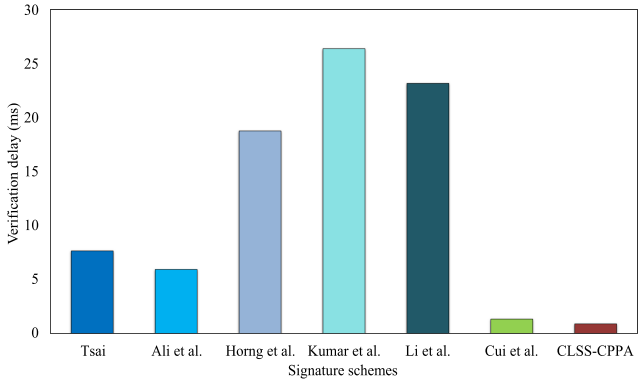


Fig. 5. Computational cost of a single signature verification.

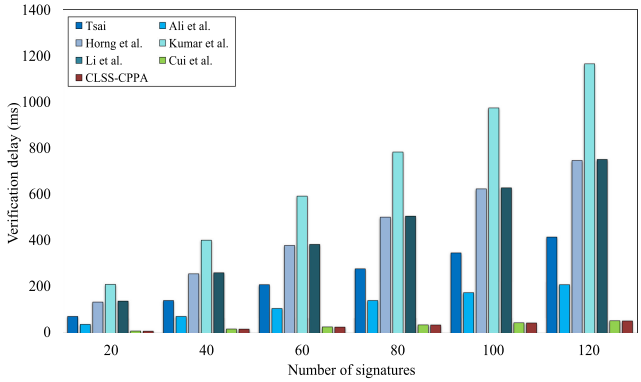


Fig. 6. Computational cost of multiple signatures verification.

TABLE III
COMPARISON OF COMPUTATIONAL COST

| Scheme | Message signing | Single signature verification | $n$ signatures verification | Pairing | Type-II security |
|---|---|---|---|---|---|
| Tsai [40] | $1M_{bp} \approx 1.709$ ms | $1P_{bp} + 2M_{bp} + 2A_{bp} \approx 7.6432$ ms | $1P_{bp} + 2nM_{bp} + 2nA_{bp} \approx 4.211 + 3.4322n$ ms | Yes | Yes |
| Ali et al. [10] | $1M_{bp} \approx 1.709$ ms | $1P_{bp} + 1M_{bp} + 1A_{bp} \approx 5.9271$ ms | $1P_{bp} + nM_{bp} + nA_{bp} \approx 4.211 + 1.7161n$ ms | Yes | Yes |
| Horng et al. [38] | $2M_{bp} + 2A_{bp} \approx 3.4322$ ms | $3P_{bp} + 1M_{bp} + 1A_{bp} + 1H_{mtp} \approx 18.7551$ ms | $3P_{bp} + nM_{bp} + nA_{bp} + nH_{mtp} \approx 12.633 + 6.1221n$ ms | Yes | No |
| Kumar et al. [41] | $4M_{bp} + 2A_{bp} + 1H_{mtp} \approx 11.2562$ ms | $4P_{bp} + 3M_{bp} + 1H_{mtp} \approx 26.377$ ms | $4P_{bp} + 3nM_{bp} + (n+1)H_{mtp} \approx 21.25 + 9.533n$ ms | Yes | Yes |
| Li et al. [39] | $2M_{bp} + 1A_{bp} + 1H_{mtp} \approx 7.8311$ ms | $3P_{bp} + 1M_{bp} + 1A_{bp} + 2H_{mtp} \approx 23.1611$ ms | $3P_{bp} + nM_{bp} + nA_{bp} + (n+1)H_{mtp} \approx 17.039 + 6.1221n$ ms | Yes | Yes |
| Cui et al. [9] | $1M_{ecc} \approx 0.442$ ms | $3M_{ecc} + 2A_{ecc} \approx 1.3296$ ms | $(n+2)M_{ecc} + 2nA_{ecc} \approx 0.844 + 0.4456n$ ms | No | No |
| CLSS-CPPA | $1M_{ecc} \approx 0.442$ ms | $2M_{ecc} + 1A_{ecc} \approx 0.8858$ ms | $(n+1)M_{ecc} + nA_{ecc} \approx 0.442 + 0.4438n$ ms | No | Yes |

TABLE IV
IMPROVEMENT OF THE CLSS-CPPA SCHEME IN PERCENTAGE

| Scheme | Msg. sign | Single sig. verify | $n$ sigs. verify |
|---|---|---|---|
| Tsai [40] | 74.14% | 88.41% | 87.09% |
| Ali et al. [10] | 74.14% | 85.06% | 74.45% |
| Horng et al. [38] | 87.12% | 95.28% | 92.81% |
| Kumar et al. [41] | 96.07% | 96.64% | 95.39% |
| Li et al. [39] | 94.36% | 96.18% | 92.86% |
| Cui et al. [9] | 0% | 33.38% | 1.21% |

TABLE V
COMPARISON OF COMMUNICATION COST

| Scheme | Single sig. transmit | $n$ sigs. transmit |
|---|---|---|
| Ali et al. [10] | $4\|\mathbb{G}_1\| + \|\mathbb{Z}_q^*\| + 4$ =536 bytes | $536n$ bytes |
| Horng et al. [38] | $4\|\mathbb{G}_1\| + \|\mathbb{Z}_q^*\| + 4$ =536 bytes | $536n$ bytes |
| Kumar et al. [41] | $4\|\mathbb{G}_1\| + \|\mathbb{Z}_q^*\| + 4$ =536 bytes | $536n$ bytes |
| Li et al. [39] | $4\|\mathbb{G}_1\| + \|\mathbb{Z}_q^*\| + 4$ =536 bytes | $536n$ bytes |
| Cui et al. [9] | $3\|\mathbb{G}\| + 2\|\mathbb{Z}_q^*\| + 4$ =164 bytes | $164n$ bytes |
| CLSS-CPPA | $3\|\mathbb{G}\| + \|\mathbb{Z}_q^*\| + 4$ =144 bytes | $144n$ bytes |

computational cost in terms of the message signing and multiple signatures verification as compared to the scheme in [9]. But our scheme is efficient in term of a single signature verification than the scheme [9]. In addition, the scheme [9] does not ensure existential unforgeability against type-II attack in the ROM as mentioned in [42]. Therefore, the CLSS-CPPA scheme efficiently and securely authenticates a single message and multiple messages that are broadcast in V2V communication.

### *B. Analysis of Communication Cost*

In this section, the communication cost of the CLSS-CPPA scheme and the related schemes in [9], [10], [38], [39], [41] are obtained and compared. We adopt the method in [25] to analyze the communication cost. We consider that status of a safety message is same in our scheme and the related schemes. According to the 80 bits security level,

The size of $\mathbb{G}_1$'s element and size of $\mathbb{G}$'s element are $64 * 2 = 128$ bytes (1024 bits) and $20 * 2 = 40$ bytes (320 bits), respectively. In [25], 20 bytes and 4 bytes are assumed to be the size of a general hash function and a time-stamp. Here, we consider the cost of an anonymous-identity $AID_i$, a public key $pk_i$, a current time-stamp $t_i$ and a signature $\Theta_i$ in each scheme in bytes.

A vehicle $\mathcal{V}_i$ in Ali *et al.* [10] transmits an anonymous-identity $AID_i = \{AID_{i,1}, AID_{i,2}\}$ for $AID_{i,1} \in \mathbb{G}_1$ and $AID_{i,2} \in \mathbb{Z}_q^*$, a public key $pk_i = \{X_i, Y_i\} \in \mathbb{G}_1$, time-stamp $t_i$, and a signature $\Theta_i \in \mathbb{G}_1$ to the vehicle $\mathcal{V}_j$. Therefore, the total communication cost generated from Ali *et al.*' scheme [10] is approximately equal to $4 * 128 + 20 + 4 = 536$ bytes. In Cui *et al.*' scheme [9], a vehicle $\mathcal{V}_i$ transmits $AID_i = \{AID_{i,1}, AID_{i,2}\}$ for $AID_{i,1} \in \mathbb{G}$ and $AID_{i,2} \in \mathbb{Z}_q^*$, $pk_i \in \mathbb{G}$, $t_i$, time-stamp $t_i$, and a $\Theta_i = \{A_i, v_i\}$, where $A_i \in \mathbb{G}$ and $v_i \in \mathbb{Z}_q^*$ to the vehicle $\mathcal{V}_j$. Therefore, Cui *et al.*' scheme [9] incurs a total communication cost, which is approximately equal to $3 * 40 + 2 * 20 + 4 = 164$ bytes. For the schemes in [38], [39], [41], the total communication costs are analyzed in the same manner and are listed in Table V. In our scheme, a vehicle $\mathcal{V}_i$ transmits $AID_i = \{AID_{i,1}, AID_{i,2}\}$ where $AID_{i,1} \in \mathbb{G}$ and $AID_{i,2} \in \{0,1\}^\pi$ (where $\pi$ indicates fixed bits' number), $pk_i \in \mathbb{G}$, $t_i$, and $\Theta_i = \{\eta_i, A_i\}$, where $\eta_i \in \mathbb{Z}_q^*$ and $A_i \in \mathbb{G}$ to the vehicle $\mathcal{V}_j$. Therefore, the total communication cost incurred from our scheme is approximately equal to $3 * 40 + 20 + 4 = 144$ bytes. From the Table V, we observe that the bilinear pairing-based schemes in [10], [38], [39], [41] generate higher cost than the ECC-based scheme in [9] and our scheme. Furthermore, the CLSS-CPPA scheme incurs low cost as compared to the scheme proposed in [9]. Hence, our scheme also improves the performance of V2V communication with respect to communication cost. Therefore, it performs better in bandwidth limited infrastructure such as VANETs.

## VIII. CONCLUSION

In this paper, an efficient and provably secure CLSS-CPPA scheme based on the ECC for V2V communication in VANETs has been proposed. We used general hash functions because they are computationally efficient as compared to the map-to-point hash functions. In addition, we used the batch signature verification method, which speeds up the performance of a receiver vehicle by simultaneously verifying multiple signatures in the environments, where multiple messages are transmitted from multiple vehicles. The CLSS-CPPA scheme ensured security with respect to the EUF-CMA against type-I and type-II attackers under the hardness assumption of the ECDLP in the ROM. We evaluated the performance of our scheme in terms of both computational cost and communication cost, which presents significant improvement when compared to the recent related schemes.

## REFERENCES

[1] Y. Liu, C. Wang, J. Huang, J. Sun, and W. Zhang, "Novel 3-D non-stationary mmWave massive MIMO channel models for 5G high-speed train wireless communications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2077–2086, Mar. 2019.

[2] M. G. Kibria, K. Nguyen, G. P. Villardi, W.-S. Liao, K. Ishizu, and F. Kojima, "A stochastic geometry analysis of multiconnectivity in heterogeneous wireless networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9734–9746, Oct. 2018.

[3] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

[4] X. Wang, T. Wei, L. Kong, L. He, F. Wu, and G. Chen, "ECASS: Edge computing based auxiliary sensing system for self-driving vehicles," *J. Syst. Architecture*, vol. 97, pp. 258–268, 2019.
[5] M. Gonzalez-Martín, M. Sepulcre, R. Molina-Masegosa, and J. Gozalvez, "Analytical models of the performance of C-V2X mode 4 vehicular communications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1155–1166, Feb. 2019.
[6] N. H. T. S. Administration, Vehicle-to-Vehicle Communication, United States Department of Transportation. Accessed: Sep. 18, 2020, [Online]. Available: https://www.nhtsa.gov/technology-innovation/vehicle-vehicle-communication
[7] B. Palaniswamy, S. Camtepe, E. Foo, and J. Pieprzyk, "An efficient authentication scheme for intra-vehicular controller area network," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 3107–3122, Mar. 2020.
[8] I. Ali, A. Hassan, and F. Li, "Authentication and privacy schemes for vehicular ad hoc networks (VANETs): A survey," *Veh. Commun.*, vol. 16, pp. 45–61, 2019.
[9] J. Cui, J. Zhang, H. Zhong, R. Shi, and Y. Xu, "An efficient certificateless aggregate signature without pairings for vehicular ad hoc networks," *Inf. Sci.*, vol. 451/452, pp. 1–15, 2018.
[10] I. Ali, M. Gervais, E. Ahene, and F. Li, "A blockchain-based certificateless public key signature scheme for vehicle-to-infrastructure communication in VANETs," *J. Syst. Architecture*, vol. 99, 2019, Art. no. 101636.
[11] K. Rabieh, M. M. Mahmoud, and M. Younis, "Privacy-preserving route reporting schemes for traffic management systems," *IEEE Trans. Veh. Technol.*, vol. 66, no. 3, pp. 2703–2713, Mar. 2017.
[12] J. Cui, J. Wen, S. Han, and H. Zhong, "Efficient privacy-preserving scheme for real-time location data in vehicular ad-hoc network," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3491–3498, Oct. 2018.
[13] L. Zhang, "OTIBAAGKA: A new security tool for cryptographic mix-zone establishment in vehicular ad hoc networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 12, pp. 2998–3010, Dec. 2017.
[14] I. Ali, T. Lawrence, and F. Li, "An efficient identity-based signature scheme without bilinear pairing for vehicle-to-vehicle communication in VANETs," *J. Syst. Architecture*, vol. 103, 2020, Art. no. 101692.
[15] L. Chen, Z. Cheng, and N. P. Smart, "Identity-based key agreement protocols from pairings," *Int. J. Inf. Secur.*, vol. 6, no. 4, pp. 213–241, 2007.
[16] V. S. Miller, "Use of elliptic curves in cryptography," in *Proc. Conf. Theory Appl. Cryptolgraph. Techn.*, 1985, pp. 417–426.
[17] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *J. Comput. Secur.*, vol. 15, no. 1, pp. 39–68, 2007.
[18] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications," in *Proc. IEEE INFOCOM-27th Conf. Comput. Commun.*, 2008, pp. 1229–1237.
[19] A. Wasef and X. Shen, "EMAP: Expedite message authentication protocol for vehicular ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 1, pp. 78–89, Jan. 2013.
[20] P. Cincilla, O. Hicham, and B. Charles, "Vehicular PKI scalability-consistency trade-offs in large scale distributed scenarios," in *Proc. IEEE Veh. Netw. Conf.*, Dec. 2016, pp. 1–8.
[21] M. Azees, P. Vijayakumar, and L. J. Deboarh, "EAAP: Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2467–2476, Sep. 2017.
[22] M. Asghar, R. R. M. Doss, and L. Pan, "A scalable and efficient PKI based authentication protocol for VANETs," in *Proc. 28th Int. Telecommun. Netw. Appl. Conf.*, Nov. 2018, pp. 1–3.
[23] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Workshop Theory Appl. Cryptolgraph. Techn.*, 1984, pp. 47–53.
[24] K.-A. Shim, "CPAS: An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks," *IEEE Trans. Veh. Technol.*, vol. 61, no. 4, pp. 1874–1883, May 2012.
[25] 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. Inf. Forensics Secur.*, vol. 10, no. 12, pp. 2681–2691, Dec. 2015.
[26] N.-W. Lo and J.-L. Tsai, "An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks without pairings," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 5, pp. 1319–1328, May 2016.
[27] Z. Jianhong, X. Min, and L. Liying, "On the security of a secure batch verification with group testing for VANET," *Int. J. Netw. Secur.*, vol. 16, no. 5, pp. 351–358, 2014.
[28] M. Bayat, M. Barmshoory, M. Rahimi, and M. R. Aref, "A secure authentication scheme for VANETs with batch verification," *Wireless Netw.*, vol. 21, no. 5, pp. 1733–1743, 2015.
[29] C.-C. Lee and Y.-M. Lai, "Toward a secure batch verification with group testing for VANET," *Wireless Netw.*, vol. 19, no. 6, pp. 1441–1449, 2013.
[30] J. K. Liu, T. H. Yuen, M. H. Au, and W. Susilo, "Improvements on an authentication scheme for vehicular sensor networks," *Expert Syst. Appl.*, vol. 41, no. 5, pp. 2559–2564, 2014.
[31] J. Cui, J. Zhang, H. Zhong, and Y. Xu, "SPACF: A secure privacy-preserving authentication scheme for VANET with Cuckoo filter," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 10 283–10 295, Nov. 2017.
[32] I. Ali and F. Li, "An efficient conditional privacy-preserving authentication scheme for vehicle-to-infrastructure communication in VANETs," *Veh. Commun.*, vol. 22, p. 100228, 2020.
[33] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, 2003, pp. 452–473.
[34] H. Xiong, Z. Guan, Z. Chen, and F. Li, "An efficient certificateless aggregate signature with constant pairing computations," *Inf. Sci.*, vol. 219, pp. 225–235, 2013.
[35] D. He, M. Tian, and J. Chen, "Insecurity of an efficient certificateless aggregate signature with constant pairing computations," *Inf. Sci.*, vol. 268, pp. 458–462, 2014.
[36] A. Malip, S.-L. Ng, and Q. Li, "Certificateless anonymous authenticated announcement scheme in vehicular Ad Hoc networks," *Secur. Commun. Netw.*, vol. 7, no. 3, pp. 588–601, 2014.
[37] A. K. Malhi and S. Batra, "An efficient certificateless aggregate signature scheme for vehicular ad-hoc networks," *Discrete Math. Theor. Comput. Sci.*, vol. 17, no. 1, pp. 317–338, 2015.
[38] S.-J. Horng, S.-F. Tzeng, P.-H. Huang, X. Wang, T. Li, and M. K. Khan, "An efficient certificateless aggregate signature with conditional privacy-preserving for vehicular sensor networks," *Inf. Sci.*, vol. 317, pp. 48–66, 2015.
[39] J. Li, H. Yuan, and Y. Zhang, "Cryptanalysis and improvement of certificateless aggregate signature with conditional privacy-preserving for vehicular sensor networks," *Networks*, vol. 317, pp. 48–66, 2015.
[40] J.-L. Tsai, "A new efficient certificateless short signature scheme using bilinear pairings," *IEEE Syst. J.*, vol. 11, no. 4, pp. 2395–2402, Dec. 2017.
[41] P. Kumar, S. Kumari, V. Sharma, X. Li, A. K. Sangaiah, and S. H. Islam, "Secure CLS and CL-AS schemes designed for VANETs," *J. Supercomput.*, vol. 75, no. 6, pp. 3076–3098, 2019.
[42] I. A. Kamil and S. O. Ogundoyin, "An improved certificateless aggregate signature scheme without bilinear pairings for vehicular ad hoc networks," *J. Inf. Secur. Appl.*, vol. 44, pp. 184–200, 2019.
[43] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *J. Cryptol.*, vol. 13, no. 3, pp. 361–396, 2000.

**Ikram Ali** received the master's degree in computer science from the Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Islamabad, Pakistan, in 2013 and the Ph.D. degree in computer science and technology from the University of Electronic Science and Technology of China, Chengdu, China, in 2020. He is currently a Postdoctoral Fellow with the School of Automation Engineering, University of Electronic Science and Technology of China. His current research interests include cryptography and network security.

**Yong Chen** (Senior Member, IEEE) was born in Sichuan, China, in 1977. He received the B.S. degree in industrial automation from the Taiyuan University of Science and Technology, Taiyuan, China, in 2001, the M.S. degree in control theory and control engineering from Guangxi University, Nanning, China, in 2004, and the Ph.D degree in control theory and control engineering from Chongqing University, Chongqing, China, in 2007. Since 2015, he has been a Professor and the Ph.D. Supervisor with the School of Automation Engineering, and the Director of the Institute of Electric Vehicle Driving System and Safety Technology, University of Electronic Science and Technology of China, Chengdu, China. He was a Visiting Scholar with the School of Mechanical Engineering, University of Adelaide, Adelaide, SA, Australia. He is currently presiding over one National Natural Science Foundation of China project, National Key Research and Development Plan Programs of China, and the Scientific and Technical Supporting Programs of Sichuan Province. He has authored or coauthored more than 80 technical papers in journals, and 20 Chinese patents. His current research interests include fault-tolerant control, network control, and intelligent connected vehicle.

**Niamat Ullah** received the master's degree in computer science from Quaid-e-Azam University, Islamabad, Pakistan, in 1996 and the Ph.D degree in information and communication engineering from Inha University, Incheon, South Korea, in 2013. From 2002 to 2016, he was an Assistant Professor with Higher Education Department, Buner, Pakistan. In 2016, he joined as an Associate Professor with Abdul Wali Khan University (Buner Campus), Mardan, Pakistan. He is currently an Associate Professor of computer science with Buner University, Buner, Pakistan. His research interests include analysis and design of MAC Protocols for WLAN, WPAN, and WBAN. He was the recipient of the Dean's Award in recognition of his excellent academic results and superior research for the Ph.D degree from the Graduate School of IT and Telecommunications, Inha University.

**Rajesh Kumar** was born in Sindh, Pakistan, in November 1991. He received the B.S. and M.S. degrees in computer science from the University of Sindh, Jamshoro, Pakistan, and the Ph.D. degree in computer science and technology from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, where he is currently working toward the Postdoctor degree in information security from the School of Computer and Engineering. He has authored or coauthored more than 20 articles in various international journals and conference proceedings. His research interests include machine learning, deep leaning, malware detection, Internet of Things (IoT), and blockchain technology.

**Wen He** was born in Sichuan, China, in 1983. He received the B.S. degree from Xihua University, Chengdu, China, in 2003 and the M.S. degree from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 2011. He is currently working toward the Ph.D. degree with the University of Electronic Science and Technology of China, Chengdu, China. He is currently a Senior Engineer. He is also the Vice Manager with Intelligent Research Institute, Chongqing Changan Automobile Company Ltd. His current research interests include V2V, safety technology, and intelligent connected vehicle.