Secure and Lightweight Conditional Privacy-Preserving Authentication for Fog-Based Vehicular Ad-Hoc Networks

Hong Zhong, Lei Chen, Jie Cui, Jing Zhang, Irina Bolodurina, Lu Liu

***Abstract*—Vehicular Ad-Hoc Networks (VANETs) play an ever-increasing important role in improving traffic management and enhancing driving safety. However, vehicular communication using a wireless channel faces security and privacy challenges. The Conditional Privacy-Preserving Authentication (CPPA) scheme is suitable for solving the above challenges, but the existing identity-based CPPA schemes suffer from inborn key escrow issues. Motivated by this, we propose a lightweight CPPA scheme based on elliptic curve cryptography to solve the above issues, in which the pseudonym and public/private key pair of the vehicle is generated by itself, so that the proposed scheme avoids the key escrow issue. Furthermore, to achieve efficient vehicular communication, a CPPA scheme is proposed using a fog computing model that supports mobility, low latency, and location awareness. The pseudonym of the vehicle is generated by two hash chains in the proposed scheme, so that the storage overhead can be reduced efficiently under the condition that backward security is guaranteed. Security analysis shows that the scheme is secure under the random oracle and satisfies the security requirements of VANETs. Performance evaluation demonstrates that the proposed scheme outperforms related schemes in terms of computational and communication overhead.**

***Index Terms*—VANETs, authentication, conditional privacy-preserving, fog computing.**

I. Introduction

VEHICULAR Ad-Hoc Networks (VANETs) are the core components of the intelligent transportation, which is vital for improving traffic management and ensuring driving safety. In VANETs, a vehicle equipped with an on-board unit (OBU) can communicate with neighboring vehicles or infrastructures. The two main communication types of VANETs are vehicle-to-vehicle and vehicle-to-infrastructure [1], which can be implemented with IEEE 802.11p or cellular networks (such as C-V2X) [2]. According to the dedicated short-range communications standard [3], the OBU regularly broadcasts safety-related information such as position, speed, steering angle, and rate of acceleration every 100-300 ms [4], which can be used to improve traffic management efficiency and reduce traffic accidents. Additionally, VANETs can provide drivers with value-added services, such as entertainment, games, uploading or downloading data through the Internet [5].

Although VANETs have a wide range of applications and provide great benefits, they also present many challenges that must be overcome. The importance of secure and private communications cannot be ignored because they can affect personal and property security [6]. Without a secure protection mechanism, messages transmitted in VANETs over the channel, which is open and insecure, can be tampered with, replayed, intercepted, or deleted by an attacker [7]. Therefore, the message authentication mechanism needs to be designed to ensure message integrity [8], [9]. Many identity-based and certificateless-based message authentication schemes have been proposed. However, the transmission delays associated with these schemes can be long and may fail to balance efficiency and safety, making it difficult to deploy these schemes in real scenarios.

Fog computing, as a new potential solution to improve the latency issue in cloud computing-based architectures [10], can be integrated into VANETs to decrease the transmission delay, provide reliable location awareness, and support access for more vehicles [11]. However, existing fog-based schemes do not provide for the protection of the privacy of vehicles. [12], [13]. For example, if a vehicle communicates with other vehicles or infrastructures using its real identity, the attacker can trace the vehicle information by monitoring the communication channel over the air. Once the driving route of the vehicle is leaked, the personal privacy of the driver will be compromised, and life safety may be threatened in the worst case [14]. Therefore, vehicular communication that uses an anonymous identity is needed. It should be emphasized that anonymity should be conditional, no one except for the trusted authority (TA) can trace an abnormally behaving vehicle [15]. The characteristic that hides the identity of a vehicle from everyone except for the TA is called conditional privacy-preserving.

A. Our Contributions

In this paper, we propose a lightweight Conditional Privacy-Preserving Authentication (CPPA) scheme based on fog com-

The work was supported in part by the National Natural Science Foundation of China under Grant 62011530046, Grant 61872001, and Grant U1936220, in part by the Russian Foundation for Basic Research under Grant 20-57-53019, and in part by the Special Fund for Key Program of Science and Technology of Anhui Province, China under Grant 202003A05020043.

H. Zhong, L. Chen, J. Cui, and J. Zhang are with the Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230039, China, the Institute of Physical Science and Information Technology, Anhui University, Hefei 230039, China (e-mail: cuijie@mail.ustc.edu.cn), and the Anhui Engineering Laboratory of IoT Security Technologies, Anhui University, Hefei 230039, China.

I. Bolodurina is with the Faculty of Mathematics and Information Technologies, Orenburg State University, Orenburg, 460018, Russia (e-mail: prmat@mail.osu.ru).

L. Liu is with the School of Informatics, University of Leicester, LE1 7RH, UK (email: l.liu@leicester.ac.uk).

puting, which retains the advantages of identity-based public key cryptography. In the proposed scheme, the vehicle can generate pseudonyms and public/private key pairs by itself. The vehicle passes requests to the TA or fog nodes (FNs) for a token that can prove the legality of their identity. The token is a short signature of the TA or FNs to the vehicle's pseudonym and public key, and can be implicitly used as the signature. During the signature sending process, the vehicle no longer needs the assistance of the server or infrastructure, so that efficient communication can be achieved. The main contributions of this study are summarized as follows.

- The vehicle can generate pseudonyms and public/private key pairs by itself, so the scheme can function without a key escrow issue. With the help of tokens supplemented by FNs for vehicles, efficient authentication between vehicles can be achieved. In addition, the workload of the TA is effectively reduced because it does not need to participate in the authentication process online.
- The hash seeds are allocated by the TA during the vehicle registration stage. The vehicle can generate pseudonyms by using two hash chains. The scheme therefore enables the anonymity of vehicles to be preserved and avoids the problem that vehicles need to pre-store a large number of pseudonyms. In addition, the scheme achieves a fast revocation of malicious vehicles because the TA only needs to release two hash seeds of the vehicle to revoke all its unexpired pseudonyms.
- We provide a formal security model to demonstrate that our scheme is provably secure. Security analysis shows that the scheme satisfies the security and privacy requirements of VANETs and resists common attacks. Performance evaluation shows that it outperforms related schemes in terms of overhead.

B. Structure of the Rest Paper

In section II, we introduce the related works of this study. The system model and security goals of fog-based VANETs is introduced in Section III. Section IV introduces preliminary knowledge related to this paper. Section V shows the authentication process in detail. Section VI and VII gives the formal security proof and analysis on the aspects of security and performance respectively. In section VIII, we provide a conclusion.

II. Related Work

Researchers have proposed the public key infrastructure-based (PKI-based) cryptography [16–18], and identity-based public key cryptography (ID-based PKC) [19–22] and certificateless-based (CL-based PKC) conditional privacy-preserving authentication (CPPA) schemes [23–26] to protect the vehicle's privacy while ensuring secure communications in VANETs [27].

Raya *et al.* [16] proposed a PKI-based scheme that uses anonymous certificates to meet conditional privacy-preserving. In their scheme, the key pairs and the corresponding anonymous certificate of vehicles are generated by the TA. The vehicle chooses a key pair to sign message for each communication. Then the verifier uses an anonymous certificate to implement message authentication. The scheme has the following weaknesses: the TA and vehicles both need to have enough storage space for storing the key pairs and corresponding anonymous certificates of all vehicles; And a malicious vehicle is revoked by the TA needs to take 1,025,430 bytes of communication overhead [28].

In 1984, Shamir *et al.* [29] first proposed the scheme of ID-based PKC, which can avoid the above certificate management problem. The vehicle's public key is composed of its public identity information, so the vehicle's identity and public key can be bound together without any certificate. However, the private key generator (PKG) can impersonate any user or decrypt any ciphertext [20]. Because all vehicles' private keys are generated by the PKG, it implies the so-called key escrow issue [30]. He *et al.* [21] uses the Schnorr signature technology [31] to propose an ID-based CPPA scheme. And the computational and communication overhead is extremely low than pairing-based bilinear schemes. However, He *et al.*'s [21] scheme depends on an ideal tamper-proof device (TPD) equipped in vehicles [9]. The scheme is hard to guarantee the security of the whole system [32], because the attacker may launch the side-channel attack (e.g., power analysis, laser scanning and so on.) to get the system private key stored in the vehicle's TPD [33], [34].

Al-Riyami *et al.* [23] proposed a CL-based PKC scheme that can solve the key escrow issue above. In their scheme, the vehicle's private key including two parts, one part is generated by the PKG and the other is generated by itself. Even if there is an attacker in collusion with PKG, the complete vehicle's private key cannot be obtained, so the key escrow issue is solved. Gong *et al.* [24] proposed a pairing-based certificateless scheme. Later, Zhang *et al.* [35] demonstrated the flaws of the security model in Gong *et al.* [24], and proposed a new CL-based PKC scheme that was proven secure. However, the scheme is not efficient because it involves many computationally complex pairing operations. Gayathri *et al.* [36] proposed a CL-based PKC scheme without pairing-based bilinear. They claimed that their scheme improves the efficiency of authentication and decreases computational overhead. However, Liu *et al.*'s [37] showed that the scheme [36] has a serious safety problem and proved it is hard against two kinds of attacks. And they claimed that they proposed a new scheme to avoid security issues in the scheme [36]. Unfortunately, Zhan *et al.* [38] claimed that Liu *et al.*'s [37] scheme fails to resist forgery attacks initiated by attackers. However, due to the lack of authentication of public keys, the CL-based PKC schemes are vulnerable to public key replacement attacks [39].

III. BACKGROUND

The fog-based VANETs system model and the detailed security goals are introduced in this section.

A. System Model

Fig. 1 illustrates the system model of this paper, it mainly includes three entities, i.e., the trust authority (TA), the fog

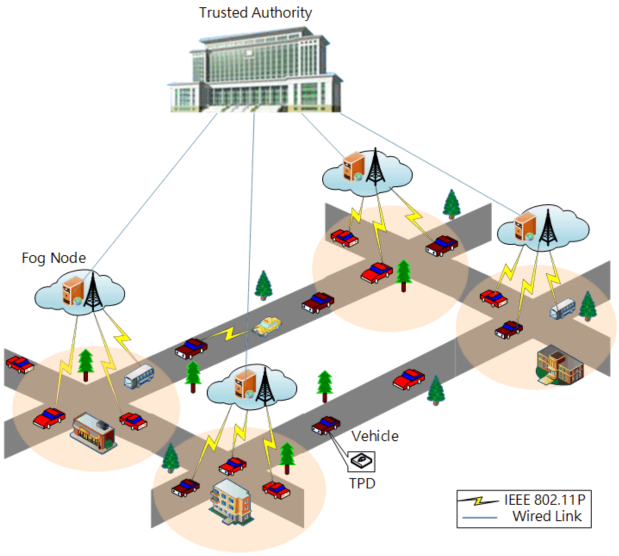


Fig. 1. The system model of the fog-based VANETs.

node (FN), and mobile vehicles.

- TA: It has sufficient computational and communication capabilities, and is responsible for initializing the whole system and providing registration services for vehicles. The TA divides its domain into several fog areas, and sends secure parameters to the FN in the fog area through a secure and private channel. No one except for the TA can trace an abnormally behaving vehicle. The TA is completely trusted worthy and will not be compromised by the attacker.
- FN: It consists of local data storage servers and wireless communication facilities, such as roadside units. To reduce the storage overhead at the vehicle side, the FN provides real-time generation of tokens services for vehicles. FNs are managed by the TA, and the TA will check all FNs periodically.
- Vehicle: The TPD equipped with each vehicle is responsible for storing the secret information and generating pseudonyms and key pairs. Each vehicle can obtain accurate time information and has reliable positioning.

B. Security Goal

The following security goals should be met for a message authentication scheme.

- ***Anonymity***: To protect the vehicle's privacy, no entity except for the TA knows the real identity by analyzing the intercepted messages.
- ***Message Authentication***: Message authentication guarantees the receiver can check the legitimacy of the message. Besides, it ensures the receiver can detect a message modified by attackers.
- ***Traceability and Identity Revocation***: When a malicious vehicle uses an anonymous identity sending messages, the TA can extract its real identity from messages and revoke it.
- ***Unlinkability***: No entity can successfully link the messages sent by the same vehicle.
- ***Backward Security***: Ensure that the leakage of the current key will not affect the subsequent keys, that is, it is not necessary to revoke the current key system and rebuild a new key system every time a key leakage is detected. The malicious vehicle has been revoked, and it should not disclose any relevant information that will reduce the conditional anonymity of the vehicle for a while before the revoking takes effect.
- ***Key Escrow Resilience***: The vehicle's complete private key is only known to itself. No one else can fake the vehicle.
- ***Common Attacks***: A basic scheme needs to resist the following types of attacks.
  1) ***Modification attack***: The attacker modifies a message and sends it to other entities in VANETs.
  2) ***Forgery attack***: The attacker forges the legitimate vehicle's signature and sends it to other entities in VANETs.
  3) ***Man-in-the-middle attack***: The adversary cannot intercept the messages over the air to perform the active attack.
  4) ***Replay attack***: Replay attack is a passive network attack, in which the attacker eavesdrops and intercepts the interactive information in the network, and resends the deliberately delayed message, thus interfering with the network communication.

IV. PRELIMINARIES

The secure hash chains [40] and elliptic curve cryptography (ECC) [31] are introduced first, which are used in the proposed scheme. Then, we formally define the scheme, which is used in Section VI.

A. Hash Chain

The one-way hash function $h(\cdot)$ has the following properties.

1) $h(.)$ takes the message of any length as input and produces a fixed-length message digests as output;
2) Known x, it is easy to compute $y = h(x)$. In contrast, known y, it is difficult to compute $x = h^{-1}(y)$;
3) Known x_1, it is computationally hard to compute $x_1 \neq x_2$ such that $h(x_1) = h(x_2)$.

The hash chain of length L is shown in Fig. 2, assuming that the initial seed value is SD and $S_i = h^i(SD), i \in [1, L]$, $h\left(h^{i-1}(x)\right) = h^i(x)$. Known S_i, it is easy to compute $S_{i+1} = h(S_i)$ but hard to get S_{i-1}.

$$SD \xrightarrow{h(.)} S_1 \xrightarrow{h(.)} S_2 \longrightarrow \cdots\cdots\ S_{i-2} \xrightarrow{h(.)} S_{i-1} \xrightarrow{h(.)} S_i$$

Fig. 2. Hash chain.

B. Elliptic Curve Cryptography

ECC construction encryption requires only a short key, so it has been widely used. Let $\mathbb{F}_p$ be a finite and $p > 3$ be a large prime number. In $\mathbb{F}_p$, $E_{a,b} : y^2 = x^3 + ax + b \ (mod\, p)$, where $a, b \in \mathbb{F}_p$ are constant numbers and $4a^3 + 27b^2 \neq 0$.

Elliptic Curve Discrete Logarithm Problem (ECDLP): The generator of the group $\mathbb{G}$ is P, given $Y = s \cdot P \in \mathbb{G}$, it is hard to know s from Y in a polynomial time.

C. The Key Escrow

The Key Escrow: The trusted authority knows the private key of each user, and it can sign documents on behalf of any user. This is called the key escrow issue.

D. The Formal Definition

In the proposed scheme, mainly four entities are engaged: the TA, FNs, vehicle, and verifier. And the formal definitions are described below.

Definition 1. The proposed scheme includes five algorithms, i.e, system initialization algorithm $Setup(\cdot)$ executed by the TA, user key generation algorithm $VKeyGen(\cdot)$ executed by the vehicle, token generation algorithm $Token(\cdot)$ executed by the TA or FNs, signature algorithm $Sign(\cdot)$ executed by the vehicle, and verification algorithm $Verify(\cdot)$ that executed by the verifier.

- $Setup(\cdot)$: It is a probabilistic algorithm that takes a security parameter λ as input, and generates the TA's master key s and public parameters $Para$ as output.
- $VKeyGen(\cdot)$: It is a probabilistic algorithm that takes public parameters $Para$ as input, and generates the vehicle's $PK_{V_{i,j}}$ and a secret key $x_{V_{i,j}}$ as output.
- $Token(\cdot)$: It is a probabilistic algorithm that takes $(Para, s, P_{pub}$, $PID_{V_{i,j}}$, $PK_{V_{i,j}}$, $TS_j)$ as input, where $PID_{V_{i,j}}$ is pseudonym of the vehicle, and generates token for $(PID_{V_{i,j}}$, $PK_{V_{i,j}})$ as output.
- $Sign(\cdot)$: It is a probabilistic algorithm that takes ($Para$, P_{pub}, $PID_{V_{i,j}}$, $PK_{V_{i,j}}$, TS_j, $token_{V_{i,j}}$, $x_{V_{i,j}}$, m_i) as input, and generates a signature σ_i as output.
- $Verify(\cdot)$: It is a deterministic algorithm that takes (σ_i, m_i) as input, and generates either "pass" which denotes a valid message, or the special symbol $\perp$ which denotes invalid message as output.

V. The Proposed Scheme

The scheme is divided into the following several phases, i.e., system setup, registration, pseudonym generation and key pairs generation, token generation, signature and verification. The TA initializes the system during the system setup phase. In the registration phase, each vehicle registers with at the TA. In the pseudonym and private key generation phase, each vehicle generates pseudonyms and key pairs by itself. In the token generation phase, the vehicle requests multiple tokens from the TA or FNs to be used for communications. In the signature and verification phase, each message needs to be signed before being broadcast to neighboring infrastructure or vehicles, then the receiver verifies the receiving messages. The proposed scheme has the following advantages: 1) Different from the traditional authentication scheme, each vehicle will generate pseudonyms and public/private key pairs by itself, so the scheme will not have the key escrow issue; 2) It provides the service of the token supplement by FNs for vehicles, which reduces the storage pressure on the vehicle; 3) It achieves fast revocation of malicious vehicles, because it just releases two hash seeds of the vehicle to revoke all its unexpired pseudonyms. Table I lists the notations.

TABLE I
EXECUTION TIME

Notations	Definitions
h_0, h_1, h_2	one-way secure hash functions
P_{pub}	The system's public key
$PID_{V_{i,j}}$	The pseudonym of V_i at TS_j
$PK_{V_{i,j}}$	The public key of V_i at TS_j
RID_i	The real identity of vehicle
s	The system's private key
$Seed_i$	The seed of pseudonym generation
tt_i	The latest timestamp
ΔT	The pseudonym's validity period
TS_j	The j-th time slot
TA	Trusted authority
V_i	The i-th vehicle
$\|\|$	The concatenation operation

A. System Setup

Let $\mathbb{F}_p$ represent the finite field, and p is a large prime number that represents the size of the finite field. The TA chooses an elliptic curve E which is defined by $y^2 = x^3 + ax + b \ (mod\ p)$, where $a, b \in \mathbb{F}_p$. $\mathbb{G}$ is the additive elliptic curve group with prime order q and generator P. O denotes infinity and $P \neq O$. Based on the public parameters $(p, q, \mathbb{G}, E, P)$, the TA initializes the system by the following steps.

1) It randomly chooses a number $s \in \mathbb{Z}_q^*$ as the system's private key, and computers $P_{pub} = s \cdot P$ as public key.
2) It chooses three one-way hash functions, $h_0 : \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$, $h_1 : \mathbb{G} \times \mathbb{G} \times \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$, $h_2 : \{0,1\}^* \times \mathbb{G} \times \mathbb{G} \times \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$.
3) It divides its domain into different fog areas. Then, the TA chooses the appropriate FN for each fog area, and transmits system's private key to the FN through secure channels.

The TA keeps the system's private key s secret, and broadcasts system parameters $Para = (p, q, P, \mathbb{G}, P_{pub}, h_1, h_2)$.

B. Registration

In this subsection, the TA is responsible for the registration of vehicles. Each vehicle submits its information to the TA, and stores the hash seeds allocated by the TA in the TPD. Last, the TA stores $(RID_i, Seed_1, Seed_2)$ into its database (DB), and stores $(Para, Seed_1, Seed_2)$ into the vehicle's TPD.

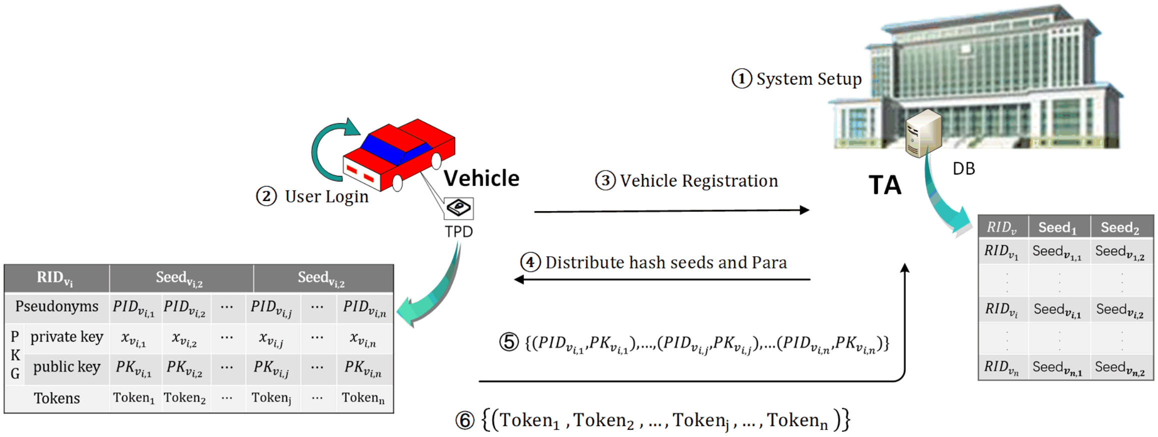

① System Setup
TA
DB
② User Login
Vehicle
TPD
③ Vehicle Registration
④ Distribute hash seeds and Para
⑤ $\{(PID_{v_{i,1}},PK_{v_{i,1}}),\dots,(PID_{v_{i,j}},PK_{v_{i,j}}),\dots(PID_{v_{i,n}},PK_{v_{i,n}})\}$
⑥ $\{(Token_1, Token_2, \dots, Token_j, \dots, Token_n)\}$
RID_{v_i} | $Seed_{v_{i,2}}$ | $Seed_{v_{i,2}}$
Pseudonyms | $PID_{v_{i,1}}$ | $PID_{v_{i,2}}$ | ⋯ | $PID_{v_{i,j}}$ | ⋯ | $PID_{v_{i,n}}$
PKG private key | $x_{v_{i,1}}$ | $x_{v_{i,2}}$ | ⋯ | $x_{v_{i,j}}$ | ⋯ | $x_{v_{i,n}}$
public key | $PK_{v_{i,1}}$ | $PK_{v_{i,2}}$ | ⋯ | $PK_{v_{i,j}}$ | ⋯ | $PK_{v_{i,n}}$
Tokens | $Token_1$ | $Token_2$ | ⋯ | $Token_j$ | ⋯ | $Token_n$
RID_v | $Seed_1$ | $Seed_2$
RID_{v_1} | $Seed_{v_{1,1}}$ | $Seed_{v_{1,2}}$
RID_{v_i} | $Seed_{v_{i,1}}$ | $Seed_{v_{i,2}}$
RID_{v_n} | $Seed_{v_{n,1}}$ | $Seed_{v_{n,2}}$

Fig. 3. The process by which the vehicle obtains the token from the TA.

C. Pseudonyms and Private Key Generation

Vehicles will generate pseudonyms and corresponding public and private key pairs by their TPD. To further improve the safety of vehicles, the system supports it to apply for a new hash seed to the TA when one of the following two conditions is met, that is, the annual inspection time is coming or the hash seed is no longer secure.

- ***Pseudonyms Generation:*** The time has been divided into w time slots, so TS_j denotes the time interval $[j * \Delta T, (j+1) * \Delta T]$, $PID_{V_{i,j}}$ is V_i's pseudonym at the time slot $TS_j (j \in [1, w])$, which is generated by two hash chains with random seeds $SD_i, 1$ and $SD_i, 2$.

$$\begin{cases} S_{1,j} = h_0^j(SD_{i,1}) \\ S_{2,w-j+1} = h_0^{w-j+1}(SD_{i,2}) \\ PID_{V_{i,j}} = h_0(S_{1,j} \oplus S_{2,w-j+1}). \end{cases} \quad (1)$$

  The proposed scheme uses the one-way hash chain technology to generate pseudonyms. Once the vehicle behaves abnormally, the TA just release two hash seeds to revoke all unexpired pseudonyms, while revoking illegal vehicles without leaking backward security [28].
- ***Keys Generation:*** The vehicle $V_{i,j}$ generates a corresponding key pair based on the pseudonym. The vehicle chooses $x_{V_{i,j}} \in \mathbb{Z}_q^*$ as the private key, and calculates $PK_{V_{i,j}} = x_{V_{i,j}} \cdot P$ as the public key.

D. Token Generation

In the proposed scheme, the vehicle can gain multiple tokens from the TA or FNs. The details are described below.

- ***Tokens from the TA:*** The interaction process by which the vehicle obtains the token from the TA during the registration phase is shown in Fig. 3. The TA will do the following to generate multiple tokens for the vehicle using the system's private key s.
  1) It chooses $r_i \in \mathbb{Z}_q^*$, calculates $C_i = r_i + s \cdot h_1(PID_{V_{i,j}} || PK_{V_{i,j}})$ and sets $A_i = r_i \cdot P$.
  2) It sends $token_{V_{i,j}} = (A_i, C_i)$ to the vehicle.
  3) The token's validity can be verified by $C_i \cdot P = A_i + h_1(PID_{V_{i,j}} || PK_{V_{i,j}}) \cdot P_{pub}$.

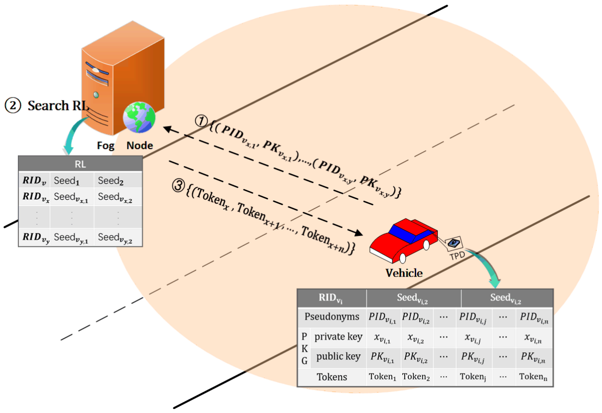

② Search RL
Fog Node
① $\{(PID_{v_{x,1}}, PK_{v_{x,1}}),\dots,(PID_{v_{x,y}}, PK_{v_{x,y}})\}$
③ $\{(Token_x, Token_{x+1}, \dots, Token_{x+n})\}$
Vehicle
TPD
RL
RID_v | $Seed_1$ | $Seed_2$
RID_{v_x} | $Seed_{v_{x,1}}$ | $Seed_{v_{x,2}}$
RID_{v_y} | $Seed_{v_{y,1}}$ | $Seed_{v_{y,2}}$
RID_{v_i} | $Seed_{v_{i,2}}$ | $Seed_{v_{i,2}}$
Pseudonyms | $PID_{v_{i,1}}$ | $PID_{v_{i,2}}$ | ⋯ | $PID_{v_{i,j}}$ | ⋯ | $PID_{v_{i,n}}$
PKG private key | $x_{v_{i,1}}$ | $x_{v_{i,2}}$ | ⋯ | $x_{v_{i,j}}$ | ⋯ | $x_{v_{i,n}}$
public key | $PK_{v_{i,1}}$ | $PK_{v_{i,2}}$ | ⋯ | $PK_{v_{i,j}}$ | ⋯ | $PK_{v_{i,n}}$
Tokens | $Token_1$ | $Token_2$ | ⋯ | $Token_j$ | ⋯ | $Token_n$

Fig. 4. The process by which the vehicle obtains the token from the FNs.

- ***Tokens from the FN:*** The interaction process between the vehicle and the FN in the fog area are shown in Fig. 4. When the FN receives a request from a vehicle, it will first check whether there is a vehicle in the revocation list (RL). If it is, the FN will reject the request from the vehicle. Otherwise, the FN uses the system's private key s distributed to it by the TA to perform the following steps to generate tokens for the vehicles.
  1) It chooses $d_i \in \mathbb{Z}_q^*$, calculates $C_i = d_i + s \cdot h_1(PID_{V_{i,j}} || PK_{V_{i,j}})$ and sets $A_i = d_i \cdot P$.
  2) It sends $token_{V_{i,j}} = (A_i, C_i)$ to the vehicle.
  3) The token's validity can be verified by $C_i \cdot P = A_i + h_1(PID_{V_{i,j}} || PK_{V_{i,j}}) \cdot P_{pub}$.

E. Message Signature

The vehicle $V_{i,j}$ generates the signature for message m_i before broadcasting it to nearby infrastructures or vehicles in VANETs.

- ***Message Signature***: The vehicle $V_{i,j}$ signs $m_i \in \{0,1\}^*$ and outputs signature $\sigma_i = (Q_i, Z_i)$ by performing the following steps.
  1) It needs to choose $b_i \in \mathbb{Z}_q^*$, computes $Q_i = A_i + b_i \cdot P$.

2) It computes $f_i = h_2(m_i||PID_{V_{i,j}}||PK_{V_{i,j}}||tt_i)$, where $tt_i \in [TS_{j-1} \cdot \Delta t, TS_j \cdot \Delta t]$ is the latest timestamp, TS_j is j-th time slot.
3) Last, it computes $Z_i = b_i + C_i + x_{V_{i,j}} \cdot f_i$, where $x_{V_{i,j}}$ is the vehicle's private key.

F. Message Verification

The legality of messages can be verified in this phase. When receiving the message $msg_i = (m_i, PID_{V_{i,j}}, PK_{V_{i,j}}, tt_i, \sigma_i)$, where $\sigma_i = (Q_i, Z_i)$, the vehicle or infrastructure verifies a signature σ_i as below.

- ***Single Verification of One Message***.
  1) The verifier first checks the validity of tt_i, if not, drops the message.
  2) The verifier finds whether $PID_{V_{i,j}}$ is in the RL, if not, the verifier continues the following steps.
  3) The verifier computes $h_{i,j} = h_1(PID_{V_{i,j}}||PK_{Vi,j})$ and $h^*_{i,j} = h_2(m_i||PID_{V_{i,j}}||PK_{V_{i,j}}||tt_i)$.
  4) Last, the verifier checks $Z_i \cdot P = Q_i + h_{i,j} \cdot P_{pub} + h^*_{i,j} \cdot PK_{V_{i,j}}$.
- ***Batch Verification***: For n distinct message-signature tuples, the verifier needs to perform the following steps to verify multiple messages, to achieve the purpose of improving the authentication efficiency [18]. The small exponent test technology is used to resist man-in-the-middle attacks in batch verification. [41]. Specifically, the verifier randomly selects $o = (o_1, o_2, ..., o_n), o_i \in [1, 2^L]$, and L is a small integer. If equation (2) holds, it will accept the message; otherwise, it will reject the message.

$$\begin{aligned}&(\Sigma_{i=1}^n o_i \cdot \ Z_i) \cdot P\\&= (\Sigma_{i=1}^n o_i \cdot Q_i) + (\Sigma_{i=1}^n o_i \cdot h_{i,j}) \cdot P_{pub}\\&\quad + (\Sigma_{i=1}^n o_i \cdot h^*_{i,j} \cdot PK_{V_{i,j}}).\end{aligned} \tag{2}$$

Next, correctness of the batch verification is presented by the following step.

$$\begin{aligned}&(\Sigma_{i=1}^n o_i \cdot \ Z_i) \cdot P\\&= (\Sigma_{i=1}^n o_i \cdot (b_i + C_i + x_{i,j} \cdot f_i)) \cdot P\\&= (\Sigma_{i=1}^n o_i \cdot b_i) \cdot P + (\Sigma_{i=1}^n o_i \cdot (a_{i,j}+\\&s \cdot h_{i,j})) \cdot P + (\Sigma_{i=1}^n o_i \cdot x_{i,j} \cdot f_i) \cdot P\\&= (\Sigma_{i=1}^n o_i \cdot Q_i) + (\Sigma_{i=1}^n o_i \cdot h_{i,j}) \cdot P_{pub}\\&\quad + (\Sigma_{i=1}^n o_i \cdot h^*_{i,j} \cdot PK_{V_{i,j}}).\end{aligned} \tag{3}$$

VI. Security Analysis

We introduce the security model and conduct formal security proof. Through the detailed security analysis, it indicates that the proposed scheme can satisfy the required security goals.

A. Security Model

According to the capability of the adversary, we consider two adversaries.

- $\mathcal{A}_I$: The adversary plays the role of dishonest signer. Therefore, it can replace the target user's public key, but the system's private key cannot be obtained.
- $\mathcal{A}_{II}$: The adversary plays the role of a malicious FNs, but it cannot replace the public key of the target user.

The game between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$ is under the existential unforgeability against adaptively chosen message attack model. The adversary $\mathcal{A}$ can query by the following oracles.

- $Setup(\cdot)$-$Oracle$: This oracle is executed by $\mathcal{C}$ taking security parameters λ as input and the system master private key as well as public parameters $Para$ as output. $\mathcal{C}$ keeps the private key of the system s secret and forwards the public parameters to $\mathcal{A}$.
- $VKeyExtract(\cdot)$-$Oracle$: This oracle is executed by $\mathcal{C}$ taking PID as input and forwarding the output to $\mathcal{A}$.
- $PublicKeyReplace(\cdot)$-$Oracle$: This oracle is executed by $\mathcal{A}$ to replace the public key of the target user without knowing the corresponding private key.
- $Corruption(\cdot)$-$Oracle$: This oracle is executed by $\mathcal{C}$ to obtain the private key of the target user. After it checks the output of $VKeyExtract(\cdot)$-$Oracle$, and forwards the user's private key to $\mathcal{A}$.
- $Token(\cdot)$-$Oracle$: This oracle is executed by $\mathcal{C}$ to response a request from $\mathcal{A}$. $\mathcal{C}$ runs this oracle, and outputs the $token_{PID,PK_{PID}}$ to $\mathcal{A}$.
- $Sign(\cdot)$-$Oracle$: This oracle is executed by $\mathcal{C}$ to get the message signature. $\mathcal{C}$ forwards the signature to $\mathcal{A}$, meanwhile records.

Based on the types of the adversary, some or all of the oracles described above maybe are queried. Therefore, different games are defined for different types of adversary $\mathcal{A}_I$ and $\mathcal{A}_{II}$.

$Game_1$: $\mathcal{A}_I$ outputs a forged signature σ^*_i on the message $(m^*_i, PID^*_{V_{i,j}})$. If the following requirements are met, we say that $\mathcal{A}_I$ wins the game.

1) σ^*_i is a valid signature of the message m^*_i.
2) $PID_{V^*_{i,j}}$ is not queried during $Corruption(\cdot)$-$Oracle$.
3) $(PID^*_{V_{i,j}}, PK^*_{V_{i,j}})$ is not queried during $Token(\cdot)$-$Oracle$.
4) $(m^*_i,PID^*_{V_{i,j}})$ is not queried during $Sign(\cdot)$-$Oracle$.

Definition 2: A signature scheme is (t, q_m, ϵ) secure in the existential unforgeability against adaptively chosen message attack security model if there exists no $\mathcal{A}_I$ who can win the above game in time t with advantage ϵ after it has made q_m signature queries.

$Game_2$: $\mathcal{A}_{II}$ outputs a forged signature σ^*_i on the message $(m^*_i, PID^*_{V_{i,j}})$. Meanwhile, if the following requirements are met, we say that $\mathcal{A}_{II}$ wins the game.

1) σ^*_i is a valid signature of the message m^*_i.
2) $PID^*_{V_{i,j}}$ is not queried during $Corruption(\cdot)$-$Oracle$.
3) $(m^*_i, PID^*_{V_{i,j}})$ is not queried during $Sign(\cdot)$-$Oracle$.

Definition 3: A signature scheme is (t, q_m, ϵ) secure in the existential unforgeability against adaptively chosen message attack security model if there exists no $\mathcal{A}_{II}$ who can win the above game in time t with advantage ϵ after it has made q_m signature queries.

B. Security Proof

Lemma 1: Only the ECDLP assumption holds in $\mathbb{G}$, the scheme is existentially unforgeable in $Game_1$. We assume

$SUCCESS_{\mathcal{A}_I}$ denotes the probability of the success of $Game_1$. The probability to solve ECDLP is $SUCCESS_{\mathcal{A}_I} \geq \frac{1}{q_m}(1-\frac{1}{q_m})^{q_m}\epsilon$, where ϵ denotes the probability to break the proposed scheme, and q_m denotes the max number of all queries.

Proof: $\mathcal{A}_I$ wins the $Game_1$ with probability ϵ. $\mathcal{C}_I$ is a challenger, aims to solve the ECDLP ($P_{pub} = s \cdot P, P, P_{pub} \in \mathbb{G}, s \in \mathbb{Z}_q^*$) with a non-negligible probability. $\mathcal{C}_I$ tries to find s with the help of $\mathcal{A}_I$ and maintains all queries' output in various lists. $\mathcal{C}_I$ first initializes a key pair list $List_{key}$, a token list $List_{Token}$, a sign list $List_{Sign}$, two secure hash value lists $List_{h_1}$ and $List_{h_2}$ as empty. Below are the queries and their output.

1) $Setup(\cdot)$: $\mathcal{C}_I$ runs this oracle by taking a secure parameter λ as input, then it chooses a private key $s \in \mathbb{Z}_q^*$, and sets the public key $P_{pub} = s \cdot P$. The public parameters $(P, P_{pub}, q, h_1, h_2)$ are sent to $\mathcal{A}_I$. $\mathcal{C}_I$ chooses an index L satisfying $1 \leq L \leq m_{h_2}$, where m_{h_2} is the number of $h_2(.)$-$Oracle$ requests.
2) $VKeyExtract(\cdot)$-$query$: Upon receiving $\mathcal{A}_I$'s $VKeyExtract(\cdot)$ query with the user's $PID_{V_{i,j}}$ $(1 \leq j \leq m_{key})$, m_{key} is the number of queries. $\mathcal{C}_I$ first finds the tuple $\langle PID_{V_{i,j}}, PK_{V_{i,j}}, x_{V_{i,j}} \rangle$ exists in $List_{key}$, and if it exists, $\mathcal{C}_I$ returns $PK_{V_{i,j}}$ to $\mathcal{A}_I$. Otherwise, it responds as follows.
    - If $j \neq L$, $\mathcal{C}_I$ chooses a private key $x_{V_{i,j}} \in \mathbb{Z}_q^*$, and sets the public key $PK_{V_{i,j}} = P \cdot x_{V_{i,j}}$.
    - If $j = L$, $\mathcal{C}_I$ sets $PK_{V_{i,j}} = P_{pub}$.

   Last, $\mathcal{C}_I$ adds the tuple $\langle PID_{V_{i,j}}, PK_{V_{i,j}}, x_{V_{i,j}} \rangle$ into $List_{key}$ and returns $PK_{V_{i,j}}$ to $\mathcal{A}_I$.
3) $h_1(\cdot)$-$query$: Upon receiving $\mathcal{A}_I$'s $h_1(\cdot)$ query with the tuple $\langle PID_{V_{i,j}}, PK_{V_{i,j}} \rangle$, $\mathcal{C}_I$ finds whether the tuple $\langle PID_{V_{i,j}}, PK_{V_{i,j}}, L_k^1 \rangle$ exists in $List_{h_1}$, where $L_k^1 = h_1(PID_{V_{i,j}}||PK_{V_{i,j}})$. If so, $\mathcal{C}_I$ returns L_k^1 to $\mathcal{A}_I$. Otherwise, $\mathcal{C}_I$ chooses $L_k^1 \in \mathbb{Z}_q^*$ and computes $L_k^1 = h_1(PID_{V_{i,j}}||PK_{V_{i,j}})$. Last, $\mathcal{C}_I$ adds the tuple $\langle PID_{V_{i,j}}, PK_{V_{i,j}}, L_k^1 \rangle$ into $List_{h_1}$ and returns L_k^1 to $\mathcal{A}_I$.
4) $h_2(\cdot)$-$query$: Upon receiving $\mathcal{A}_I$'s $h_2(\cdot)$ query with the tuple $\langle m_i, PID_{V_{i,j}}, PK_{V_{i,j}}, tt_i \rangle$, $\mathcal{C}_I$ finds whether the tuple $\langle\ m_i, PID_{V_{i,j}},\ PK_{V_{i,j}}, tt_i, L_k^2 \rangle$ exists in $List_{h_2}$, where $L_k^2 = h_2(m_i||PID_{V_{i,j}}||PK_{V_{i,j}}||tt_i)$. If so, $\mathcal{C}_I$ returns L_k^2 to $\mathcal{A}_I$. Otherwise, $\mathcal{C}_I$ chooses $L_k^2 \in \mathbb{Z}_q^*$ and computes $L_k^2 = h_2(m_i||PID_{V_{i,j}}||PK_{V_{i,j}}||tt_i)$. Last, $\mathcal{C}_I$ adds the tuple $\langle m_i, PID_{V_{i,j}}, PK_{V_{i,j}}, tt_i, L_k^2 \rangle$ into $List_{h_2}$ and returns L_k^2 to $\mathcal{A}_I$.
5) $PublicKeyReplace(\cdot)$-$query$: Upon receiving $\mathcal{A}_I$'s $PublicKeyReplace(\cdot)$ query with the tuple $\langle PID_{V_{i,j}}, PK_{V_{i,j}}^* \rangle$, $\mathcal{C}_I$ finds whether the tuple $\langle PID_{V_{i,j}}, PK_{V_{i,j}}, x_{V_{i,j}} \rangle$ exists in $List_{key}$, and if it exists, $\mathcal{C}_I$ will replace the tuple with $\langle PID_{V_{i,j}}, PK_{V_{i,j}}^*, \perp \rangle$.
6) $Corruption(\cdot)$-$query$: Upon receiving query form $\mathcal{A}_I$, $\mathcal{C}_I$ first finds whether the tuple $\langle PID_{V_{i,j}}, PK_{V_{i,j}}, x_{V_{i,j}} \rangle$ exists in $List_{key}$. If so, $\mathcal{C}_I$ returns $x_{V_{i,j}}$ to $\mathcal{A}_I$. Otherwise, $\mathcal{C}_I$ chooses the private key $x_{V_{i,j}} \in \mathbb{Z}_q^*$ and sets $PK_{V_{i,j}} = x_{V_{i,j}} \cdot P$. Last, $\mathcal{C}_I$ returns $x_{V_{i,j}}$ to $\mathcal{A}_I$ and updates $List_{key}$.
7) $Token(\cdot)$-$query$: Upon receiving a query from $\mathcal{A}_I$, $\mathcal{C}_I$ takes the following steps.
    - If $j \neq L$, $\mathcal{C}_I$ first finds whether $token_{V_{i,j}} = (A_i, C_i)$ exists in $List_{Token}$. If so, $\mathcal{C}_I$ returns $token_{V_{i,j}}$ to $\mathcal{A}_I$. Otherwise, $\mathcal{C}_I$ chooses $C_i, L_k^1 \in \mathbb{Z}_p^*$ and computes $Q_i = C_i \cdot P - L_k^1 \cdot P_{pub}$. Then, $\mathcal{C}_I$ finds $List_{h_1}$ to check whether $(PID_{V_{i,j}}, PK_{V_{i,j}})$ has already been defined. If found, $\mathcal{C}_I$ again chooses $C_i, L_k^1 \in \mathbb{Z}_q^*$ until there is no collision. Last, $\mathcal{C}_I$ updates the $List_{h_1}$ and $List_{Token}$, and outputs $token_{V_{i,j}} = (A_i, C_i)$.
    - If $j = L$, $\mathcal{C}_I$ aborts the game.
8) $Sign(\cdot)$-$query$: Upon receiving a query from $\mathcal{A}_I$, $\mathcal{C}_I$ runs above oracles to get ($PID_{V_{i,j}}$,$PK_{V_{i,j}}$,$x_{V_{i,j}}$). If $x_{V_{i,j}} = \perp$, $\mathcal{A}_I$ is required returning $x_{V_{i,j}}$ to $\mathcal{C}_I$. Otherwise, $\mathcal{C}_I$ will perform the following steps.
    - If $j \neq L$, $\mathcal{C}_I$ runs $Token(\cdot)$ as input to get token. Then, $\mathcal{C}_I$ computes signature on $(m_i, PID_{V_{i,j}})$ by using $token_{V_{i,j}}$ and $x_{V_{i,j}}$.
    - If $j = L$, $\mathcal{C}_I$ chooses $L_k^1, L_k^2, Q_j, Z_j \in \mathbb{Z}_q^*$, $Q_j = Z_j - L_k^1 \cdot P_{pub} - L_k^2 \cdot PK_{V_{i,j}}$. $L_k^1 = h_1(PID_{V_{i,j}}||PK_{V_{i,j}})$, $L_k^2 = h_2(m_i||PID_{V_{i,j}}||PK_{V_{i,j}}||tt_i)$. If the hash values collide, it will rechoose the parameters and performs again. It is responsible for updating the $List_{h_1}$ and $List_{h_2}$. Last, $\mathcal{C}_I$ returns (Q_i, Z_i) as a signature to $\mathcal{A}_I$.

Analysis: Through using the Forking lemma [42], $\mathcal{A}_I$ generates two signatures $\sigma = (Q_i, Z_i)$ and $\sigma^* = (Q_i, Z_i^*)$ on the message with the same $PK_{V_{i,j}}$, but $h_{i,j} \neq h'_{i,j}$.

$$Z_i \cdot P = Q_i + h_{i,j} \cdot P_{pub} + h_{i,j}^* \cdot P_{pub}. \tag{4}$$

$$Z_i^* \cdot P = Q_i + h'_{i,j} \cdot P_{pub} + h_{i,j}^* \cdot P_{pub}. \tag{5}$$

According to the above equations (4) and (5), $\mathcal{C}_I$ outputs $s = (Z_i - Z_i^*) \cdot (h_{i,j} - h'_{i,j})^{-1}$ as a solution for the ECDLP. With the above game, the success probability of $\mathcal{C}_I$ solves the ECDLP based on the following events.

a) Event E_1: $\mathcal{C}_I$ does not quit.
b) Event E_2: $\mathcal{A}_I$ can forge a valid signature.
c) Event E_3: $\mathcal{A}_I$ can forge the identity of the target user.

Therefore, $Pr[E_1] \geq (1 - 1/q_m)^{q_m}$, $Pr[E_2|E_1] = \epsilon$ and $Pr[E_3|E_2 \Lambda E_1] = (1/q_m)$, we obtain $SUCCESS_{\mathcal{A}_I} \geq \frac{1}{q_m}(1-\frac{1}{q_m})^{q_m}\epsilon$. The scheme is secure against forgery under the adaptive chosen message attack in the random oracle model, because the ECDLP is hard to solve in polynomial time.

Lemma 2: The scheme is existentially unforgeable in $Game_2$. We assume $SUCCESS_{\mathcal{A}_{II}}$ denotes the probability of the success of $Game_2$. If the ECDLP assumption holds in $\mathbb{G}$, the probability to solve ECDLP is $SUCCESS_{\mathcal{A}_{II}} \geq \frac{1}{q_m}(1-\frac{1}{q_m})^{q_m}\epsilon$, where ϵ denotes the probability to break the proposed scheme, and q_m denotes the max number of all queries.

Proof: $\mathcal{A}_{II}$ wins the $Game_2$ with probability ϵ. $\mathcal{C}_{II}$ is challenger, aims to solve ECDLP $(P, P_{pub} = s \cdot P, P, P_{pub} \in \mathbb{G}, s \in \mathbb{Z}_q^*)$ with a non-negligible probability. $\mathcal{C}_{II}$ maintains all queries' output in various lists. $\mathcal{C}_{II}$ first initializes a key pair list $List_{key}$, a token list $List_{Token}$, a sign list $List_{Sign}$, two secure hash value lists $List_{h_1}$ and $List_{h_2}$ as empty. Below are the queries and their output.

1) $Setup(\cdot)$: $\mathcal{C}_{II}$ runs this oracle by taking a secure parameter λ as input, then it randomly selects $s \in \mathbb{Z}_q^*$ as the system's private key and sets $P_{pub} = s \cdot P$ as the public key. Next, $(P, P_{pub}, q, h_1, h_2)$ as the system's public parameters. $\mathcal{C}_{II}$ chooses an index L satisfying $1 \leq L \leq m_{h_2}$, where m_{h_2} is the number of $h_2(.) - Oracle$ query.
2) $VKeyExtract(\cdot)$-$query$: Upon receiving $\mathcal{A}_{II}$'s $VKeyExtract(\cdot)$ query with the user's $PID_{V_{i,j}}(1 \leq j \leq m_{key})$, and m_{key} denotes the number of the key queries. $\mathcal{C}_{II}$ first finds whether the tuple $\langle PID_{V_{i,j}}, PK_{V_{i,j}}, x_{V_{i,j}} \rangle$ exists in $List_{key}$, and if it exists, $\mathcal{C}_{II}$ returns $PK_{V_{i,j}}$ to $\mathcal{A}_{II}$. Otherwise, it responses as follows.
   - If $j \neq L$, $\mathcal{C}_{II}$ chooses $x_{V_{i,j}} \in \mathbb{Z}_q^*$ and sets $PK_{V_{i,j}} = P \cdot x_{V_{i,j}}$.
   - If $j = L$, $\mathcal{C}_{II}$ sets $PK_{V_{i,j}} = P_{pub}$.

   Last, $\mathcal{C}_{II}$ adds the tuple $\langle PID_{V_{i,j}}, PK_{V_{i,j}}, x_{V_{i,j}} \rangle$ into the $List_{key}$ and returns $PK_{V_{i,j}}$ to $\mathcal{A}_{II}$.
3) $h_1(\cdot)$-$query$: Upon receiving $\mathcal{A}_{II}$'s $h_1(\cdot)$ query with the tuple $\langle PID_{V_{i,j}}, PK_{V_{i,j}} \rangle$, $\mathcal{C}_{II}$ checks whether the tuple $\langle PID_{V_{i,j}}, PK_{V_{i,j}}, L_k^1 \rangle$ exists in $List_{h_1}$, where $L_k^1 = h_1(PID_{V_{i,j}} || PK_{V_{i,j}})$. If so, $\mathcal{C}_{II}$ returns L_k^1 to $\mathcal{A}_{II}$. Otherwise, $\mathcal{C}_{II}$ chooses $L_k^1 \in \mathbb{Z}_q^*$ and computes $L_k^1 = h_1(PID_{V_{i,j}} || PK_{V_{i,j}})$. Last, $\mathcal{C}_{II}$ adds $(PID_{V_{i,j}}, PK_{V_{i,j}}, L_k^1)$ into $List_{h_1}$ and returns L_k^1 to $\mathcal{A}_{II}$.
4) $h_2(\cdot)$-$query$: Upon receiving $\mathcal{A}_{II}$'s $h_2(\cdot)$ query with the tuple $\langle m_i, PID_{V_{i,j}}, PK_{V_{i,j}}, tt_i \rangle$, $\mathcal{C}_{II}$ checks whether the tuple $\langle m_i, PID_{V_{i,j}}, PK_{V_{i,j}}, tt_i, L_k^2 \rangle$ exists in $List_{h_2}$, where L_k^2 = $h_2(m_i || PID_{V_{i,j}} || PK_{V_{i,j}} || tt_i)$. If so, $\mathcal{C}_{II}$ returns L_k^2 to $\mathcal{A}_{II}$. Otherwise, $\mathcal{C}_{II}$ chooses $L_k^2 \in \mathbb{Z}_q^*$ and computes L_k^2 = $h_2(m_i || PID_{V_{i,j}} || PK_{V_{i,j}} || tt_i)$. Last, $\mathcal{C}_{II}$ adds $(m_i, PID_{V_{i,j}}, PK_{V_{i,j}}, tt_i, L_k^2)$ into $List_{h_2}$ and returns L_k^2 to $\mathcal{A}_{II}$.
5) $Corruption(\cdot)$-$query$: Upon receiving a query from $\mathcal{A}_{II}$, $\mathcal{C}_{II}$ takes the following steps.
   - If $j \neq L$, $\mathcal{C}_{II}$ checks whether the tuple $\langle PID_{V_{i,j}}, PK_{V_{i,j}}, x_{V_{i,j}} \rangle$ exists in $List_{key}$. If so, $\mathcal{C}_{II}$ returns $x_{V_{i,j}}$ to $\mathcal{A}_{II}$. Otherwise, $\mathcal{C}_{II}$ chooses $x_{V_{i,j}} \in \mathbb{Z}_q^*$ and sets $PK_{V_{i,j}} = x_{V_{i,j}} \cdot P$. Then, $\mathcal{C}_{II}$ returns $x_{V_{i,j}}$ to $\mathcal{A}_{II}$ and updates $List_{key}$.
   - If $j = L$, $\mathcal{C}_{II}$ returns $\perp$.
6) $Sign(\cdot)$-$query$: Upon receiving a query from $\mathcal{A}_{II}$, $\mathcal{C}_{II}$ takes the following steps.
   - If $j \neq L$, $\mathcal{C}_{II}$ runs $Token(\cdot)$ with $(PID_{V_{i,j}}, PK_{V_{i,j}})$, executes $Sign(\cdot)$, and returns the signature to $\mathcal{A}_{II}$.
   - If $j = L$, $\mathcal{C}_{II}$ chooses L_k^1, L_k^2, Q_j, $Z_j \in \mathbb{Z}_q^*$, $Q_j = Z_j - L_k^1 \cdot P_{pub} - L_k^2 \cdot PK_{V_{i,j}}$, where $L_k^1 = h_1(PID_{V_{i,j}} || PK_{V_{i,j}})$, and $L_k^2 = h_2(m_i || PID_{V_{i,j}} || PK_{V_{i,j}} || tt_i)$. If hash values collide, it rechooses the values and performs it again. It updates the $List_{h_1}$ and $List_{h_2}$. Last, $\mathcal{C}_{II}$ returns (Q_i, Z_i) as a signature to $\mathcal{A}_{II}$.

Analysis: Through applying the Forking lemma [42], $\mathcal{A}_{II}$ can generate two signatures $\sigma = (Q_i, Z_i)$ and $\sigma^* = (Q_i, Z_i^*)$ on the message with the same $PK_{V_{i,j}}$, but $h_{i,j}^*$ and $(h_{i,j}^*)'$ are not equal.

$$Z_i \cdot P = Q_i + h_{i,j} \cdot P_{pub} + h_{i,j}^* \cdot P_{pub}. \qquad (6)$$

$$Z_i^* \cdot P = Q_i + h_{i,j} \cdot P_{pub} + (h_{i,j}^*)' \cdot P_{pub}. \qquad (7)$$

According to the above equations (6) and (7), $\mathcal{C}_{II}$ outputs $x_{V_{i,j}} = (Z_i - Z_i^*) \cdot (h_{i,j} - (h_{i,j}^*)')^{-1}$ as a solution for the ECDLP. With the above game, $\mathcal{C}_{II}$ can solve the ECDLP based on the following events.

a) Event E_1: $\mathcal{C}_{II}$ does not quit.
b) Event E_2: $\mathcal{A}_{II}$ can forge a valid signature.
c) Event E_3: $\mathcal{A}_{II}$ can forge the identity of the target user.

Therefore, $Pr[E_1] \geq (1 - 1/q_m)^{q_m}$,$Pr[E_2|E_1] = \epsilon$ and $Pr[E_3|E_2 \Lambda E_1] = (1/q_m)$, we obtain $SUCCESS_{\mathcal{A}_{II}} \geq \frac{1}{q_m}(1 - \frac{1}{q_m})^{q_m}\epsilon$. The scheme is secure against forgery under the adaptive chosen message attack in the random oracle model because the ECDLP is hard to solve in polynomial time.

C. Security Analysis

- ***Anonymity***: The vehicle's real identity is hidden in a pseudonym $PID_{V_{i,j}}$, therefore, the proposed scheme can achieve identity's anonymity.
- ***Message Authentication***: According to security proof in the previous section, it is generally known that as long as the ECDLP is computation hard, attackers can not forge a valid signature. Therefore, the receiver can verify the message's integrity by verifying whether the equation $Z_i \cdot P = Q_i + h_{i,j} \cdot P_{pub} + h_{i,j}^* \cdot PK_{V_{i,j}}$ holds. Thus, the scheme can meet the requirements of message authentication.
- ***Traceability and Identity Revocation***: In the proposed scheme, the TA releases two hash seeds of the malicious vehicle to revoke all its unexpired pseudonyms.
- ***Unlinkability***: The vehicle frequently updates its pseudonym to ensure security, and only the TA knows the vehicle's true identity and the relationship between the two pseudonyms $PID_{V_{i,j}} = h_0\left(S_{1,j} \oplus S_{2,w-j+1}\right)$ and $PID_{V_{i',j+1}} = h_0\left(S_{1,j+1} \oplus S_{2,w-j}\right)$ with the knowledge of hash seeds. If the attacker aims to check their relationship, it has to compute $\alpha = h_1^{-1}\left(S_{1,j} \oplus S_{2,w-j+1}\right)$ and $\beta = h_0^{-1}(\alpha \oplus S_{1,j})$, $h_0(h_0(S_{1,j} \oplus \beta)) = PID_{V_{i',j+1}}$. For a l-bit one-way hash function, the overhead of solving h_1^{-1} is $O(2^{2l-1})$. If both pseudonyms $PID_{V_{i,j}}$ and $PID_{V_{i',j-1}}$ come from the same user, the relationship of h_1 needs 2^{l-1} times to verify. Thus, all overhead is $O(2^{2l-1})$.
- ***Backward Security***: In our scheme, the attacker cannot obtain the following key from the leaked password. It is impossible for any entity to obtain the pseudonym

used by the revoked vehicle. If hash seeds $S_{1,j}$ and $S_{2,w-j+1}$ are used to revoke the vehicle V_i at the time slot tt_i, the adversary needs to know $S_{1,j-1} = {h_0}^{-1}$ to compute V_i' pseudonyms in the slot TS_{j-1}, i.e., $PID_{V_{i,j-1}} = h_0\left(S_{1,j-1} \oplus S_{2,w-j+2}\right)$. Assuming a 224-bit one-way hash function, such as SHA-2, it is difficult to search $S_{1,j-1}$ from $S_{1,j}$.

- ***Common Attacks***: A basic scheme needs to resist the following types of attacks.
  1) ***Modification attack***: The message authentication is ensured by the proposed signature scheme. Once the message is modified, it cannot pass the verification.
  2) ***Forgery attack***: The attacker cannot obtain a $token_{V_{i,j}} = (A_i, C_i)$ for its pseudonym and public key, the signature generated by the attacker will not pass the equation $Z_i \cdot P = Q_i + h_{i,j} \cdot P_{pub} + h^*_{i,j} \cdot P_{pub}$.
  3) ***Man-in-the-Middle attack***: The message is secure by the proposed signature scheme, which cannot be broken in a polynomial time.
  4) ***Replay attack***: The latest timestamp tt_i is used to calculate $Z_i = b_i + C_i + x_{V_{i,j}} \cdot f_i$ and $f_i = h_1(m_i || PID_{V_{i,j}} || PK_{V_{i,j}} || tt_i)$. By checking the validity of the timestamp, replay attacks can be avoided.

VII. Performance Analysis

The contributions and deficiencies of related schemes [21], [43], [44] are demonstrated first in this section.

The scheme of He *et al.* [21] proposed the first ID-based CPPA scheme in VANETs. Their scheme abandons the use of bilinear pairing of traditional encryption techniques. Batch verification is used, which greatly decreases the computational and communication overhead. In their scheme, the TA needs to distribute the private key of the system to the vehicle's TPD through a secure channel. Thus, the pseudonyms and private keys of the vehicle are generated by its TPD. It avoids pre-storing a lot of certificates and pseudonyms in the vehicle, and greatly reduces storage costs. However, their proposed scheme relies heavily on TPD. It is dangerous to store the master private key in the TPD [45], because many studies [34] [46] have shown that the adversary can obtain sensitive data from the vehicle's TPD by launching a side-channel attack. Once the private key of the system is leaked, the entire system will be compromised [32].

The scheme of Shen *et al.* [43] used the data recovery feature method to propose a vehicle cloud security data aggregation scheme for VANETs, which can meet the security properties of data confidentiality, privacy-preserving, and resistance to replay attacks. However, Shen *et al.*'s scheme [43] has following some deficiencies, i.e., first, vehicles use their own real identity for communication, so it does not satisfy the properties of conditional privacy-preserving in VANETs; second, the scheme using the bilinear pairing cryptographic algorithm caused high computational and communication overhead; third, the public/private key pair generated by the roadside unit (RSU) for the vehicle needs to provide a secure channel for transmission. It requires a certain transmission overhead and communication resources, which limits the availability of practical usage.

The scheme of Sutrala *et al.* [44] proposed a new CPPA authentication mechanism for VANETs, in which vehicles can be authenticated by neighboring vehicles, and RSUs can perform batch verification of vehicles within their range. However, in Sutrala *et al.*'s scheme, the RSU generates pseudonyms for the vehicles in its area. When there are many vehicles requesting service in the area, it is prone to increase latency and channel congesting.

In the proposed scheme, vehicles generate key pairs, and the TA or FNs can provide a token of the vehicle's identity (pseudonym, public key), which avoids the key escrow issue in ID-based PKC. The token can implicitly be used as the signature, and transmitted to vehicles over the open channels, which greatly reduces transmission overhead. During the signature sending process, the vehicle no longer depends on the help of the server or roadside infrastructure to achieve efficient communication. More details are described below.

A. Computational Overhead Analysis and Comparison

He *et al.* [21] and Sutrala *et al.* [44] both used the ECC encryption algorithm, and Shen *et al.* [43] used the bilinear pairing encryption algorithm. To more clearly analyze the computational and communication overhead of basic cryptography operations, the cryptographic operations related to the scheme are designed as follows. $\mathbb{G}_1$ is an additive group, and a symmetric bilinear pairing $\overline{E} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$. Similarly, we construct the ECC algorithm: $\mathbb{G}$ is an additive group and a non-singular elliptic curve E. We utilize C/C++ cryptography called Miracl [47] to evaluate all the cryptographic operation's execution times. Under the ubuntu 18.04 environment with Intel i7-6700 processor, and 8GB memory. The execution time is as shown in the Table.

Next, we first introduce He *et al.*'s [21] scheme, generating a pseudonym of each vehicle is required to execute one one-way hash operation and two scalar multiplication operations about the ECC. Therefore, the time of generating a pseudonym is $2T_{ecc_{sm}} + 2T_h \approx 0.7092$ms. The signature of each vehicle needs to execute two one-way hash function operations and one scalar multiplication operation about the ECC. Thus, the execution time of the signature is $T_{ecc_{sm}} + T_h \approx 0.3546$ms. Then, to verify one single message, the vehicle executes three scalar multiplication and two point addition operations about the ECC, and two one-way hash operations. So, the time of verification is $3T_{ecc_{sm}} + 2T_{ecc_a} + 2T_h \approx 1.0670$ms. Batch verification is required to perform $(2n)$ one-way hash function, and $(n+2)$ scalar multiplication, $(3n-1)$ point addition and $(2n)$ small scalar multiplication operations about the ECC. The execute time of BVMM is $(n+2)T_{ecc_{sm}} + (3n-1)T_{ecc_a} + (2n)T_{ecc_{sm.s}} + (2n)T_h \approx (0.4346n + 0.7026)$ms.

Shen *et al.*'s scheme [43] did not hide the anonymity of the vehicle, so the time of generating a pseudonym is 0ms. The signature of each vehicle is required to execute three hash operations, one pairing and a scalar multiplication operation about the bilinear pairing. Thus, the signature's all time is $T_{bp} + T_{bp_{sm}} + T_e + 3T_h \approx 6.1640$ms. Then, to verify one

TABLE
EXECUTION TIME OF CRYPTOGRAPHIC OPERATIONS

Symbol	Description	Time(ms)
T_{bp}	The execution time of a bilinear pairing operation $e(\overline{P},\overline{Q})$,where $\overline{P},\overline{Q} \in \mathbb{G}_1(\mathbb{G}_1$ is additive group).	4.8923
$T_{bp_{sm}}$	The execution time of a scale multiplication operation $\overline{x} \cdot \overline{P}$ related to the bilinear pairing, where $x \in \mathbb{Z}_q^*$ and $\overline{P} \in \mathbb{G}_1$.	1.0052
T_{bp_a}	The execution time of a point addition operation $(\overline{P}+\overline{Q})$ related to the bilinear pairing, where $\overline{P}+\overline{Q} \in \mathbb{G}_1$.	0.0079
T_e	The execution time of Power in $\mathbb{G}_1$.	0.2605
T_{mtp}	The execution time of a hash-to-point operation related to the bilinear pairing.	2.4853
$T_{ecc_{sm}}$	The execution time of a scale multiplication operation $x \cdot P$ related to the ECC, where $P \in \mathbb{G}$ and $x \in \mathbb{Z}_q^*$.	0.3526
$T_{ecc_{sm.s}}$	The execution time of a small scale multiplication operation $o_{i,j} \cdot P$ used in the small exponent test technology, where $P \in \mathbb{G}$, $o_{i,j}$ is a small random integer in $[1, 2^t]$ and t is a small integer.	0.0351
T_{ecc_a}	The execution time of a point addition operation $P + Q$ related to the ECC, where $P, Q \in \mathbb{G}$.	0.0026
T_h	The execution time of a general hash function operation.	0.002

single message, the vehicle is required to execute three hash operations and two pairing operations. The verification needs time $2T_{bp} + T_{bp_{sm}} + T_e \approx 11.0503$ms. Batch verification is required to execute $(2n)$ bilinear pairing and n scalar multiplication operations, n power and $(2n-1)$ point addition operations. The BVMM's execution time is $(2n)T_{bp} + nT_e + nT_{bp_{sm}} + 2(n-1)T_{bp_a} \approx (11.0661n - 0.0158)$ms.

Sutrala *et al.*'s scheme [44], generating a pseudonym of each vehicle executes four scalar multiplication operations about the ECC, two hash operations. Thus, generating a pseudonym needs $4T_{ecc_{sm}} + 2T_h \approx 1.4144$ms. Next, the signature of each vehicle is required to execute one one-way hash function operation, and one scalar multiplication operation about the ECC. So, the time of the signature needs $T_{ecc_{sm}} + T_h \approx 0.3546$ms. Then, to verify one single message, each vehicle executes two hash function operations, and three scalar multiplication operations, two point addition operations about the ECC. Therefore, the time of the verification needs $3T_{ecc_{sm}} + 2T_{ecc_a} + 2T_h \approx 1.0670$ms. Batch verification of multiple messages executes $(2n)$ hash function, $(3n)$ scalar multiplication and $(4n-2)$ point addition and $(2n)$ small scalar multiplication operations about the ECC. The time of BVMM is $(3n)T_{ecc_{sm}} + (4n-2)T_{ecc_a} + (2n)T_{ecc_{sm.s}} + (2n)T_h \approx (1.1424n - 0.0052)$ms.

In the proposed scheme, generating a pseudonym of each vehicle is required to execute three hash operations and one scalar multiplication operation on ECC. Thus, the time of generating a pseudonym needs $3T_h + T_{ecc_{sm}} \approx 0.3586$ms. Then, the signature is required to execute one scalar multiplication operation and two point addition operations on ECC, and one hash function operation. Thus, the time required to signature is $T_{ecc_{sm}} + T_{ecc_a} + T_h \approx 0.3572$ms. For verifying one single message is required to perform one scalar multiplication and one point addition operation about the ECC, and one hash function operation, so the time is $3T_{ecc_{sm}} + 2T_{ecc_a} + 2T_h \approx 1.0670$ms. Batch verification of multiple messages executes $(2n)$ hash function operations, and $(n+2)$ scalar multiplication, $(3n-1)$ point addition and $(2n)$ small scalar multiplication operations about the ECC. The time of BVMM needs $(n+2)T_{ecc_{sm}} + (2n)T_{ecc_a} + (2n)T_{ecc_{sm.s}} + (2n)T_h \approx (0.432n + 0.7052)$ms.

The vehicle can generate pseudonyms by using two hash chains, and each token generation essentially amounts to one elliptic curve point multiplication, which reduces much time-consuming operations and improves the security of vehicle users and the efficiency of message authentication. As shown in Fig. 5, we can see that the proposed scheme has a relatively lower computational overhead compared with several related schemes [21], [43], [44]. And Table III shows the computational overhead of each step for several related schemes.

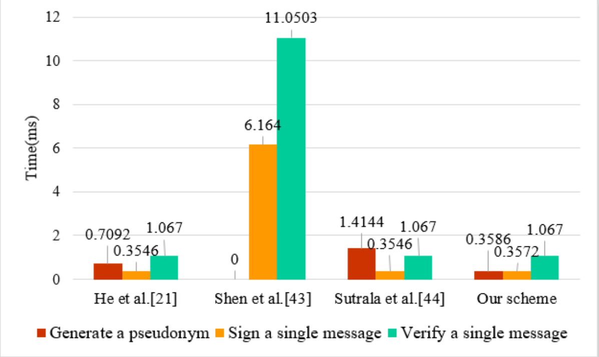


Fig. 5. Comparison Of Computational Overhead.

B. Communication Overhead Analysis

In this subsection, we have evaluated the communication overhead of the above scheme in detail. Let the sizes of the element in $\mathbb{G}$, the element in $\mathbb{G}_1$ and the element in $\mathbb{Z}_q^*$ be 40 bytes, 128 bytes, and 20 bytes, respectively. At the same

TABLE II
COMPARISON OF COMPUTATIONAL OVERHEAD

	GOP	SOSM	VOSM	BVMM
He $et\ al.$'s scheme [21]	$2T_{ecc_{sm}} + 2T_h \approx 0.7092$ms	$T_{ecc_{sm}} + T_h \approx 0.3546$ms	$3T_{ecc_{sm}} + 2T_{ecc_a} + 2T_h \approx 1.0670$ms	$(n+2)T_{ecc_{sm}} + (3n-1)T_{ecc_a} + (2n)T_{ecc_{sm.s}} + (2n)T_h \approx (0.4346n + 0.7026)$ms
Shen $et\ al.$'s scheme [43]	0ms	$T_{bp} + T_{bp_{sm}} + T_e + 3T_h \approx 6.1640$ms	$2T_{bp} + T_{bp_{sm}} + T_e \approx 11.0503$ms	$(2n)T_{bp} + nT_e + nT_{bp_{sm}} + 2(n-1)T_{bp_a} \approx 11.0661n - 0.0158$ms
Sutrala $et\ al.$'s scheme [44]	$4T_{ecc_{sm}} + 2T_h \approx 1.4144$ms	$T_{ecc_{sm}} + T_h \approx 0.3546$ms	$3T_{ecc_{sm}} + 2T_{ecc_a} + 2T_h \approx 1.0670$ms	$(3n)T_{ecc_{sm}} + (4n-2)T_{ecc_a} + (2n)T_{ecc_{sm.s}} + (2n)T_h \approx (1.1424n - 0.0052)$ms
Our scheme	$T_{ecc_{sm}} + 3T_h \approx 0.3586$ms	$T_{ecc_{sm}} + T_{ecc_a} + T_h \approx 0.3572$ms	$3T_{ecc_{sm}} + 2T_{ecc_a} + 2T_h \approx 1.0670$ms	$(n+2)T_{ecc_{sm}} + (2n)T_{ecc_a} + (2n)T_{ecc_{sm.s}} + (2n)T_h \approx (0.432n + 0.7052)$ms

GOP: generation of one pseudonym.
SOSM: signing of one single message.
VOSM: verification of one single message.
BVMM: Batch verification of multiple messages.

TABLE III
COMMUNICATION OVERHEAD

Schemes	Broadcasting a message (bytes)	Broadcasting n messages (bytes)
[21]	144	144n
[43]	428	428n
[44]	244	244n
Our	124	124n

time, we set the output of the hash function to 20 bytes, and the pseudonym and time stamp to 20 bytes and 4 bytes, respectively.

In He $et\ al.$'s scheme [21], the vehicle broadcasts the message $\{PID_i, T_i, R_i, \sigma_i\}$ to nearby vehicles or infrastructures with an anonymous identity, where PID_i is composed of $PID_{i,1}$ and $PID_{i,2}$, $PID_{i,1}, PID_{i,2}, R_i \in \mathbb{G}, \sigma_i \in \mathbb{Z}_q^*$, T_i denotes the timestamp. Therefore, the communication overhead is (40 $*$ 3 + 20 + 4) = 144 bytes. In Shen $et\ al.$'s scheme [43], the vehicle broadcasts $\{ID_i, T_i, \sigma_i$ is composed of $(e(g,g)^{r_1}, r_2, \eta)\}$ to nearby vehicles or infrastructures, where $ID_i, r_2 \in \mathbb{Z}_q^*, e(g,g)^{r_1} \in \mathbb{G}_T, \eta \in \mathbb{G}_1$. Thus, the communication overhead needs (128 + 20 $*$ 2 + 256 + 4) = 428 bytes. In Sutrala $et\ al.$'s scheme [44], the vehicle broadcasts $\{AID_i = (AID_{i,1}, AID_{i,2}),\ \sigma_i = (f_i, g_i), B_i, K_i, R_i, T_1\}$ to nearby vehicles or infrastructures, where $AID_{i,2}, f_i, g_i \in \mathbb{Z}_q^*$, and $AID_{i,1}, B_i, K_i, R_i \in \mathbb{G}$. So, the communication overhead is (40 $*$ 4 + 20 $*$ 3 + 4) = 224 bytes. In the proposed scheme, the vehicle broadcasts $\{PID_{V_{i,j}}, PK_{V_{i,j}}, tt_i, \sigma_i\}$ to nearby vehicles or infrastructures, where $\sigma_i = (Q_i, Z_i), PK_{V_{i,j}}, Q_i \in \mathbb{G}$, $Z_i \in \mathbb{Z}_q^*$ and tt_i denotes the current timestamp. Therefore, the communication overhead needs (20 + 40 + 20 + 40 + 4) = 124 bytes. From Table III, we can see that the proposed scheme compared with several related schemes [21], [43], [44] has lower communication overhead.

VIII. CONCLUSION

In this paper, we proposed a lightweight CPPA scheme for fog-based VANETs. Specifically, the pseudonym and key pair of the vehicle is generated by itself, so the key escrow issue is avoided. In VANETs, the vehicle sends requests to the TA or FNs for a token that can be used to verify the legality of the identity. The token can be transmitted over open channels without relying on private channels. For malicious vehicles, the unexpired pseudonyms of an errant user can be easily withdrawn by releasing just two hash seeds. Given the importance of vehicular communication security, we proved the security of the scheme in the random oracle model. Finally, a detailed security analysis demonstrated that the proposed scheme satisfies the security and privacy requirements required by VANETs. And the performance analysis of the computation and communication overhead verified that our scheme is more secure and efficient.

ACKNOWLEDGMENT

The authors are very grateful to the anonymous referees for their detailed comments and suggestions regarding this paper.

REFERENCES

[1] J. Cheng, J. Cheng, M. Zhou, F. Liu, S. Gao, and C. Liu, "Routing in internet of vehicles: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2339–2352, 2015.

[2] Z. H. Mir and F. Filali, "Lte and ieee 802.11 p for vehicular networking: a performance evaluation," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, p. 89, 2014.

[3] C. Cseh, "Architecture of the dedicated short-range communications (dsrc) protocol," in *VTC'98. 48th IEEE Vehicular Technology Conference. Pathway to Global Wireless Revolution (Cat. No. 98CH36151)*, vol. 3, pp. 2095–2099, IEEE, 1998.

[4] A. Boualouache, S.-M. Senouci, and S. Moussaoui, "A survey on pseudonym changing strategies for vehicular ad-hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 770–790, 2017.

[5] H. Peng, L. Liang, X. Shen, and G. Y. Li, "Vehicular communications: A network layer perspective," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1064–1078, 2018.

[6] J. Cheng, G. Yuan, M. Zhou, S. Gao, C. Liu, H. Duan, and Q. Zeng, "Accessibility analysis and modeling for iov in an urban scene," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4246–4256, 2020.

[7] J. Cui, X. Zhang, H. Zhong, J. Zhang, and L. Liu, "Extensible conditional privacy protection authentication scheme for secure vehicular networks in a multi-cloud environment," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1654–1667, 2019.

[8] X. Lin and X. Li, "Achieving efficient cooperative message authentication in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 7, pp. 3339–3348, 2013.

[9] J. Zhang, J. Cui, H. Zhong, Z. Chen, and L. Liu, "Pa-crt: Chinese remainder theorem based conditional privacy-preserving authentication scheme in vehicular ad-hoc networks," *IEEE Transactions on Dependable and Secure Computing*, 2019.

[10] O. T. T. Kim, N. D. Tri, N. H. Tran, C. S. Hong, *et al.*, "A shared parking model in vehicular network using fog and cloud environment," in *2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 321–326, IEEE, 2015.

[11] G. Sun, S. Sun, H. Yu, and M. Guizani, "Toward incentivizing fog-based privacy-preserving mobile crowdsensing in the internet of vehicles," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4128–4142, 2019.

[12] Y. Wu, J. Wu, L. Chen, G. Zhou, and J. Yan, "Fog computing model and efficient algorithms for directional vehicle mobility in vehicular network," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[13] Z. Ning, J. Huang, and X. Wang, "Vehicular fog computing: Enabling real-time traffic management for smart cities," *IEEE Wireless Communications*, vol. 26, no. 1, pp. 87–93, 2019.

[14] L. Song, G. Sun, H. Yu, X. Du, and M. Guizani, "Fbia: A fog-based identity authentication scheme for privacy preservation in internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5403–5415, 2020.

[15] M. Azees, P. Vijayakumar, and L. J. Deboarh, "Eaap: Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2467–2476, 2017.

[16] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of computer security*, vol. 15, no. 1, pp. 39–68, 2007.

[17] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "Ecpp: Efficient conditional privacy preservation protocol for secure vehicular communications," in *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*, pp. 1229–1237, IEEE, 2008.

[18] S.-J. Horng, S.-F. Tzeng, Y. Pan, P. Fan, X. Wang, T. Li, and M. K. Khan, "b-specs+: Batch verification for secure pseudonymous authentication in vanet," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 11, pp. 1860–1875, 2013.

[19] C. Zhang, R. Lu, X. Lin, P.-H. Ho, and X. Shen, "An efficient identity-based batch verification scheme for vehicular sensor networks," in *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*, pp. 246–250, IEEE, 2008.

[20] N.-W. Lo and J.-L. Tsai, "An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks without pairings," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 5, pp. 1319–1328, 2015.

[21] D. He, S. Zeadally, B. Xu, and X. Huang, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 2681–2691, 2015.

[22] J. Cui, L. Wei, J. Zhang, Y. Xu, and H. Zhong, "An efficient message-authentication scheme based on edge computing for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1621–1632, 2018.

[23] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *International conference on the theory and application of cryptology and information security*, pp. 452–473, Springer, 2003.

[24] Z. Gong, Y. Long, X. Hong, and K. Chen, "Two certificateless aggregate signatures from bilinear maps," in *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007)*, vol. 3, pp. 188–193, IEEE, 2007.

[25] 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," *Information Sciences*, vol. 317, pp. 48–66, 2015.

[26] J. Cui, D. Wu, J. Zhang, Y. Xu, and H. Zhong, "An efficient authentication scheme based on semi-trusted authority in vanets," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2972–2986, 2019.

[27] R. Hussain, J. Lee, and S. Zeadally, "Trust in vanet: A survey of current solutions and future research opportunities," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[28] S. Jiang, X. Zhu, and L. Wang, "An efficient anonymous batch authentication scheme based on hmac for vanets," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 8, pp. 2193–2204, 2016.

[29] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Workshop on the theory and application of cryptographic techniques*, pp. 47–53, Springer, 1984.

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

[31] C.-P. Schnorr, "Efficient signature generation by smart cards," *Journal of cryptology*, vol. 4, no. 3, pp. 161–174, 1991.

[32] J. Li, Y. Ji, K.-K. R. Choo, and D. Hogrefe, "Cl-cppa: certificate-less conditional privacy-preserving authentication protocol for the internet of vehicles," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10332–10343, 2019.

[33] Y. Yarom and N. Benger, "Recovering openssl ecdsa nonces using the flush+ reload cache side-channel attack.," *IACR Cryptol. ePrint Arch.*, vol. 2014, p. 140, 2014.

[34] D. Genkin, L. Pachmanov, I. Pipman, E. Tromer, and Y. Yarom, "Ecdsa key extraction from mobile devices via nonintrusive physical side channels," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1626–1638, 2016.

[35] L. Zhang and F. Zhang, "A new certificateless aggregate signature scheme," *Computer Communications*, vol. 32, no. 6, pp. 1079–1085, 2009.

[36] N. Gayathri, G. Thumbur, P. R. Kumar, M. Z. U. Rahman, P. V. Reddy, *et al.*, "Efficient and secure pairing-free certificateless aggregate signature scheme for healthcare wireless medical sensor networks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9064–9075, 2019.

[37] J. Liu, L. Wang, and Y. Yu, "Improved security of a pairing-free certificateless aggregate signature in healthcare wireless medical sensor networks," *IEEE Internet of Things Journal*, 2020.

[38] Y. Zhan, B. Wang, and R. Lu, "Cryptanalysis and improvement of a pairing-free certificateless aggregate signature in healthcare wireless medical sensor networks," *IEEE Internet of Things Journal*, 2020.

[39] X. Huang, W. Susilo, Y. Mu, and F. Zhang, "On the security of certificateless signature schemes from asiacrypt 2003," in *International Conference on Cryptology and Network Security*, pp. 13–25, Springer, 2005.

[40] W. Mao, *Modern cryptography: theory and practice*. Pearson

Education India, 2003.

[41] Z. Jianhong, X. Min, and L. Liying, “On the security of a secure batch verification with group testing for vanet,” *International Journal of Network Security*, vol. 16, no. 5, pp. 351–358, 2014.

[42] D. Pointcheval and J. Stern, “Security arguments for digital signatures and blind signatures,” *Journal of cryptology*, vol. 13, no. 3, pp. 361–396, 2000.

[43] J. Shen, D. Liu, X. Chen, J. Li, N. Kumar, and P. Vijayakumar, “Secure real-time traffic data aggregation with batch verification for vehicular cloud in vanets,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 807–817, 2019.

[44] A. K. Sutrala, P. Bagga, A. K. Das, N. Kumar, J. J. Rodrigues, and P. Lorenz, “On the design of conditional privacy preserving batch verification-based authentication scheme for internet of vehicles deployment,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5535–5548, 2020.

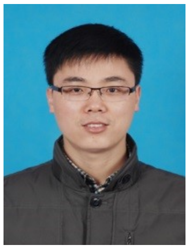
[45] K.-A. Shim, “CPAS: an efficient conditional privacy-preserving authentication scheme for vehicular sensor networks,” *IEEE Transactions on Vehicular Technology*, vol. 61, no. 4, pp. 1874–1883, 2012.

[46] S. Ravi, A. Raghunathan, and S. Chakradhar, “Tamper resistance mechanisms for secure embedded systems,” in *17th International Conference on VLSI Design. Proceedings.*, pp. 605–611, IEEE, 2004.

[47] “Miracl cryptographic sdk.” https://github.com/miracl/MIRACL/. Accessed 29 Nov, 2019.

Hong Zhong was born in Anhui Province, China, in 1965. She received her Ph.D. degree in computer science from University of Science and Technology of China in 2005. She is currently a professor and Ph.D. supervisor of the School of Computer Science and Technology at Anhui University. Her research interests include applied cryptography, IoT security, vehicular ad hoc network, cloud computing security and software-defined networking (SDN). She has over 200 scientific publications in reputable journals (e.g., IEEE Journal on Selected Areas in Communications, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Mobile Computing, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, IEEE Transactions on Intelligent Transportation Systems, IEEE Transactions on Multimedia, IEEE Transactions on Vehicular Technology, IEEE Transactions on Network and Service Management, IEEE Transactions on Cloud Computing, IEEE Transactions on Industrial Informatics, IEEE Transactions on Industrial Electronics and IEEE Transactions on Big Data), academic books and international conferences.

Lei Chen is now a research student in the School of Computer Science and Technology, Anhui University. Her research focuses on communication security in vehicular ad hoc network.



Jie Cui was born in Henan Province, China, in 1980. He received his Ph.D. degree in University of Science and Technology of China in 2012. He is currently a professor and Ph.D. supervisor of the School of Computer Science and Technology at Anhui University. His current research interests include applied cryptography, IoT security, vehicular ad hoc network, cloud computing security and software-defined networking (SDN). He has over 150 scientific publications in reputable journals (e.g., IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, IEEE Journal on Selected Areas in Communications, IEEE Transactions on Mobile Computing, IEEE Transactions on Parallel and Distributed Systems, IEEE Transactions on Computers, IEEE Transactions on Vehicular Technology, IEEE Transactions on Intelligent Transportation Systems, IEEE Transactions on Network and Service Management, IEEE Transactions on Industrial Informatics, IEEE Transactions on Industrial Electronics, IEEE Transactions on Cloud Computing and IEEE Transactions on Multimedia), academic books and international conferences.

Jing Zhang is currently a Ph.D. student in the School of Computer Science and Technology, Anhui University, Hefei, China. Her research interests include vehicular ad hoc network, IoT security and applied cryptography. She has nearly 20 scientific publications in reputable journals (e.g. IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, IEEE Journal on Selected Areas in Communications, IEEE Transactions on Vehicular Technology, IEEE Transactions on Intelligent Transportation Systems, Information Sciences, Science China Information Sciences and Vehicular Communications) and international conferences.

Irina Bolodurina is currently a professor and head of Department of Applied Mathematics, at the Orenburg State University. She received her Ph.D. degree from South Ural State University. Prof. Irina Bolodurina has over 60 scientific publications in academic journals and international conferences which indexing in Scopus and WoS. She has participated in over 20 scientific projects supported by the RFBR and other Russian scientific programs. She’s current research interests include theory of optimal control, mathematical modeling, information analysis software, control of social and economic systems, decision support systems, data integration, and processing.

Lu Liu is the Professor of Informatics and Head of School of Informatics in the University of Leicester, UK. Prof Liu received the Ph.D. degree from University of Surrey, UK and MSc in Data Communication Systems from Brunel University, UK. Prof Lius research interests are in areas of cloud computing, service computing, computer networks and peer-to-peer networking. He is a Fellow of British Computer Society (BCS).