LBVP: A Lightweight Batch Verification Protocol for Fog-Based Vehicular Networks Using Self-Certified Public Key Cryptography

Xiaoyu Zhang, Hong Zhong, Jie Cui, Irina Bolodurina, Lu Liu

***Abstract*—To achieve efficient vehicular network communication and service, researchers proposed fog-based vehicular networks (FVNs). One of the prerequisites for developing large-scale FVNs is ensuring the security and privacy of the entire network environment. However, the existing schemes proposed for FVNs exist considerable calculation and communication costs and/or security vulnerabilities. Therefore, to promote efficient FVN authentication, we propose a lightweight security protocol using self-certified public key cryptography. In the protocol, the trusted authority does not need to participate in the authentication process between the vehicle and the fog node online. And the vehicle can dynamically update its login password and pseudonym, without performing complicated interactive steps with the trusted authority. In addition, our protocol supports batch verification, which significantly improves the system authentication efficiency. A detailed security analysis reveals that our protocol can meet the security requirements of vehicular networks while resisting the common types of attack. Calculation and communication overhead comparisons further prove that our protocol exhibits better performance than related schemes.**

***Index Terms*—Self-certified, fog-based vehicular networks, batch verification, lightweight**

I. Introduction

COMMUNICATION technology, especially the 5G technology, has greatly promoted the development of vehicular networks [1]. Nowadays, vehicles are gradually being equipped with more powerful equipment, e.g., tachographs, radars, and global positioning systems. And the on-board unit (OBU) enables the vehicle to exchange information with nearby vehicles and infrastructure, thereby achieving two main applications [2]. One application involves using road-hazard reports, emergency brake warnings, and other safety-related information to effectively reduce road congestion and avoid traffic accidents, thereby improving vehicle safety. The other includes the infotainment applications that provide passengers with a comfortable and enjoyable driving experience through high-definition video streaming, peer-to-peer games, social media access, etc.

Although vehicular networks have wide-ranging applications and provide great benefits, many challenges must be overcome. Although vehicles have certain computing and storage capabilities, compared with the huge amount of data generated and collected by an intelligent vehicle in a day (about 25 GB/h), the task of storing and processing data on the vehicle is still very arduous. Introducing cloud computing to vehicular networks seems to be a viable solution to this problem [3]. But due to the increasing number of vehicles requesting cloud services, the workload of the cloud is increasingly heavy [4]. In addition, because of the distance from the service terminals, the service quality of delay-sensitive applications is even less satisfactory and can easily lead to large round-trip delays and high energy consumption [5]. In this regard, researchers proposed to build a vehicular network model based on fog computing.

The fog-based vehicular networks environment has several unique features [6]: i) low latency. With the computing power of fog nodes, messages sent by vehicles can be processed locally to achieve lower time delay; ii) location awareness. It can cache hotspot information related to the geographic location in the fog node, such as the information of nearby parking lots and gas stations, to achieve location-based services; iii) supporting more edge nodes. Distributed deployment of fog nodes can coordinate to respond to a huge number of vehicle requests. It has been proved that with the computing and communication capabilities provided by a large number of fog nodes, better vehicular applications and services can be achieved. However, because fog nodes are usually deployed in unattended environments, and vehicles interact with fog nodes through wireless networks. Therefore, it is necessary to provide secure communication protection mechanisms for resisting various network attacks [7].

Many cryptography-based schemes for fog-based vehicular networks (FVNs) have been proposed. However, these related schemes mainly have the following limitations. Firstly, some schemes do not provide satisfactory performance as they involve bilinear pairing operation. Therefore, high computation and communication costs are caused, such as in schemes [8], [9], [10], [11], and [12]. Moreover, according to dedicated short-range communications (DSRC) protocol, vehicles send a safety message every 100-300 milliseconds. Assume 10 neighboring vehicles send messages every 200 ms to an roadside units (RSU) simultaneously, the number of messages that the RSU needs to process in one minute is nearly 3000. If the RSU handles messages one by one, the problem of delay

X. Zhang, H. Zhong, J. Cui 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, and the Anhui Engineering Laboratory of IoT Security Technologies, Anhui University, Hefei 230039, China (e-mail: zhongh@ahu.edu.cn).

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).

will emerge, whereas batch verification is not considered in schemes [13] and [14]. Secondly, some schemes have security vulnerabilities. For example, the scheme in [9] cannot provide unlinkability, and scheme in [15] is unable to resist replay attacks, and scheme in [11] cannot resist impersonation attacks. Thirdly, certificate-based authentication schemes imply that the public key has to be accompanied by the certificate generated by the certification authority (CA); consequently, vehicles are required to store the public key and certificate issued by the CA, which leads to high storage costs, such as in scheme [12]. More importantly, although the existing certificateless or identity-based authentication schemes can avoid this problem, some schemes require a remote authentication server (AS) to participate in the authentication process online. And tedious interactions easily lead to larger authentication and transmission delay, such as schemes [16] and [13]. In addition, when there is only a single AS, increasing service requests are easy to cause a single point of failure and performance bottleneck, thus affecting service efficiency.

In [17], Girault first introduced self-certified public keys. Compared with the previous public key cryptography, self-certified public key cryptography avoids the problem of certificate management, and enables users to achieve authentication without the online registration center. Due to this advantage, presently, it has been applied to many different scenarios [18], [19], [20]. However, because of the unique characteristics of the vehicle to fog communication, existing self-certified public key cryptography-based authentication schemes in other fields cannot be directly applied to FVNs. Specifically, due to a large number of vehicles, frequent information interaction, and some security applications are sensitive to time delays, fast and efficient authentication is required. Moreover, because of the limited network resources and the limited computing and storage capacity of the OBU, the scheme should minimize the calculation and communication overhead. In addition, vehicular communication in a wireless network environment needs to be able to resist common types of attacks and realize conditional privacy protection. Furthermore, according to our investigation, existing self-certified public key cryptography-based schemes [21], [22] for vehicular networks are not well applicable for FVNs in terms of overhead or security.

A. Our Motivations

Considering the huge number of vehicles, frequent information interaction between vehicles and fog nodes, and limited network bandwidth and computation resources, it is necessary to propose a fast and efficient authentication scheme for FVNs. According to the above analysis, the limitations of the existing schemes designed for secure communication in FVNs can be summarized as: i) not providing satisfactory performance in computation and communication overhead; ii) there are security vulnerabilities and cannot resist common types of attacks; iii) in schemes that vehicles need to store and update certificates, there are certificate management problems; iv) fast authentication between vehicles and fog nodes is not supported. Moreover, although the self-certified public key cryptography has been proved to be able to realize that it does not require a third trusted authority to participate in the authentication process online, and has been applied to various fields. However, because of the unique characteristics and security requirements of the vehicle to fog communication, existing self-certified public key cryptography-based schemes in other fields cannot be directly applied to FVNs. To fill these research gaps, we are motivated to propose a secure and efficient authentication scheme based on self-certified public key cryptography, which supports batch verification and is suitable for the FVN.

B. Our Contributions

The main contributions of this study are summarized as follows.

- Our protocol can achieve fast and efficient authentication between large-scale vehicles and fog nodes. It effectively reduces the trusted authority (TA) workload because it does not require the TA to assist with the authentication process online. The vehicle can generate the available pseudonym by itself without sending a pseudonym update request to the TA.
- We provide a formal security model to demonstrate that our self-certified public key cryptography-based protocol is provably secure. Moreover, a detailed security analysis shows that it can meet security requirements, e.g., conditional privacy protection in vehicular networks, and resist common types of security attacks.
- The protocol is based on elliptic-curve cryptographic operations, rather than complex bilinear pairing operations, and it supports batch verification. A comparison of the computing and communication overhead with related schemes proves that it achieves better performance.

C. Outline

In Section II, we introduce the related work. Section III shows the background, including network model and assumptions, security objectives, and related preliminaries. In Section IV, we demonstrate the process of our protocol in detail. In Section V and VI, we make formal security proof and analysis and performance analysis. Finally, in Section VII, we draw conclusions.

II. Related Work

In this section, we will introduce related work from two aspects, namely authentication schemes that proposed for fog-based vehicular networks and schemes that based on self-certified public key cryptography. A comparative summary of some existing authentication schemes is presented in Table I.

A. Authentication Schemes for Fog-based Vehicular Networks

The concept of fog computing proposed by Cisco in 2012 has now been extended to the field of vehicular networks [29]. Although the entities that play the role of fog nodes in vehicular networks are different, the fog nodes in most systems are servers that extend from the cloud to the edge. And fog nodes are used to reduce the latency of reaction time and the

TABLE I
EXISTING AUTHENTICATION SCHEMES: A COMPARATIVE SUMMARY

Scheme	Year	Description	Limitations / Drawbacks
[8]	2017	*Use certificateless aggregate signcryption; *Bilinear pairing; *For road surface condition monitoring in the FVN.	*Updating the pseudonym of the scheme is not considered; *High computation cost.
[5]	2018	*For addressing location privacy issues in the FVN; *Boneh-Boyen short signature.	*Vehicles cannot generate or update valid pseudonyms independently.
[9]	2019	*For collision avoidance system in 5G fog-based IoV; *Bilinear pairing; *Use certificateless aggregate signcryption.	*Cannot provide unlinkability [23]; *High computation cost.
[13]	2019	*For fog-based vehicular ad-hoc networks authentication; *Elliptic curve cryptosystem (ECC).	*Traceability is not considered [23]; *Batch verification is not considered; *Authentication phase rely on the trusted cloud service provider.
[15]	2019	*For road condition monitoring authentication in FVN; *Certificateless aggregate signcryption.	*Traceability and identity privacy are not provided; *Unable to resist replay attacks [24].
[10]	2019	*For real-time traffic data aggregation in VANETs; *Bilinear pairing.	*High computation cost; *The signature transmission relies on a secure channel, which limits the practical deployment [25]; *Conditional privacy-preserving characteristic is not satisfied [25].
[14]	2020	*For shared data auditing in FVN; *Certificateless signcryption.	*Batch verification is not considered.
[11]	2020	*For realizing fast handover authentication and avoid a single point of failure in vehicular networks; *Bilinear pairing.	*High computation cost; *Unable to resist impersonation attacks.
[12]	2021	*For authentication of semi-trusted RSUs in VANETs; *Bilinear pairing.	*High computation cost; *Certificates need to be generated and updated for the vehicle.
[19]	2016	*For multi-server remote user authentication; *Bilinear pairing; *Use self-certified public key cryptography.	*Cannot withstand the replay attacks and denial of service attacks [26]; *High computation cost; *Batch verification is not considered.
[20]	2018	*For key distribution in the smart grid; *Elliptic curve cryptosystem (ECC); *Use self-certified public key cryptography.	*Lack anonymity [27]; *Batch verification is not considered.
[18]	2020	*For multi-server remote user authentication; *Elliptic curve cryptosystem (ECC); *Use self-certified public key cryptography.	*Cannot withstand the replay attacks [28]; *Batch verification is not considered.

cost of computing and communication [30]. Based on such an architecture, researchers have explored a variety of potential applications and proposed many corresponding authentication schemes [31], [32].

In [33], Lu *et al.* proposed an efficient conditional privacy preservation protocol for vehicular networks, where roadside units (RSUs) are in charge of issuing dynamic short-term anonymous keys for the vehicle to achieve anonymity. The scheme involved bilinear pairing operations. Later, Huang *et al.* [34] pointed out that there is a fairly high latency in the key generation process. Moreover, since the vehicles acquire their pseudonyms from the RSU, the revocation of the malicious vehicle cannot be achieved.

For enhancing security in vehicular crowdsensing-based road surface condition monitoring system, in [8], Basudan *et al.* proposed privacy-preserving certificateless aggregate signcryption protocol. The protocol support batch verification, but it involved bilinear pairing operations. Moreover, the pseudonym used by vehicles is fixed, and the protocol did not consider the dynamic updating of vehicle pseudonyms.

Later, in [5], Kang *et al.* claimed that traditional centralized pseudonym management is likely to lead to large delays and high costs, and then proposed a cloud-fog-vehicle three-tier hierarchical architecture and a privacy-preserving protocol for protecting the location privacy. The work of pseudonym management shifts to specialized fogs, but it also implies that vehicles cannot independently generate or update valid pseudonyms. Because vehicles need a large number of pseudonyms, it will lead to a large communication and storage overhead in order to update pseudonyms in time.

In 2019, Nkenyereye *et al.* [9] proposed a secure and privacy-preserving collision avoidance system for 5G fog-based IoV. The fog nodes are responsible for collecting speed violation reports (TVRs) from vehicles. The fog nodes aggregated and batch verified the signatures on the TVRs and then broadcast notifications to other entities. In their protocol, bilinear pairing operations and MapToPoint hash operation are required, therefore, it will lead to high computation cost. And Qin *et al.* [23] pointed out that their protocol did not provide the unlinkability between the TVRs.

To facilitate secure interaction in fog-based VANETs, Ma *et al.* [13] designed an authenticated key agreement protocol without bilinear pairing. In their protocol, strict formal security proof is presented. And they evaluated the efficiency of the protocol to show its practicality. But they did not consider batch verification, and the authentication between vehicle users and fog nodes has to rely on the assistance of a trusted cloud service provider.

B. Self-Certified Public Key Cryptography-based Schemes

In [17], Girault first introduced self-certified public keys. Later, He *et al.* [19] proposed an anonymous mobile user authentication protocol for protecting the rights of authorized users. However, due to the large computational overhead of bilinear pairing, the performance of their scheme is not satisfactory. And because the protocol did not have secure timestamp verification, it is vulnerable to reply attacks and denial of service attacks [26].

In 2016, Haripriya *et al.*[35] proposed an ECC-based self-certified key management scheme (SCKM) for IoT systems. And the technique of zero-knowledge proof was integrated. The authors proved their scheme is more energy-efficient than other certificate-based schemes. And in 2017, in Li *et al.*'s scheme [36], for authenticating sensitive information in smart mobile communication, they proposed a self-certified scheme that combined with the NTRU signature algorithm. Due to the application scenario, the above schemes [35] and [36] do not consider the security requirement of conditional privacy protection.

In view of the problems that existed in the key distribution scheme of the smart grid, a key distribution protocol was proposed in [20]. In this protocol, the real identity of the service provider will be stored in the smart meter during the registration phase. But Wu *et al.* [27] claimed that it lacks anonymity.

In 2020, Wang *et al.* [18] proposed a two-factor lightweight authentication protocol using self-certified public key cryptography for multi-server 5G networks. The scheme is based on lightweight elliptic curve cryptosystem, but Hossain *et al.* [28] pointed out that the scheme exists security flaws.

In 2021, Li *et al.* presented a blockchain-based public auditing protocol for cloud data using self-certified public key system [37]. The scheme involves a bilinear map and Merkle hash tree (MHT) which is an authentication structure built based on hashes of data.

However, because vehicular networks require fast and efficient authentication and need to meet security requirements in particular conditional privacy protection, the above-mentioned schemes based on self-certified cryptography cannot be well fitted to this application scenarios. The existing self-certified cryptography based schemes proposed for vehicular networks are as follows.

In [21], Rabadi *et al.* first applied self-certified encryption to vehicular networks for realizing the anonymity of drivers. But their scheme cannot satisfy the non-linkability. Rabadi *et al.* revised this problem in [38] later. And in [39], they also proposed two broadcast communication schemes for wireless access in vehicular environments. However, the schemes are not supported batch verification.

In [40], Li *et al.* proposed a secure mobile electronic payment scheme for vehicular value-added services in the restricted connectivity scenarios. It uses self-certified key agreement to establish the shared symmetric key between the vehicle and the bank without additional message exchange. However, it does not consider the user password change problem. In [41], Wang *et al.* proposed a self-certified public key protocol to support value-added vehicular services too. Their scheme supports batch authentication, but it is based on the computationally expensive bilinear pair operation. The protocol of Zhang *et al.* [42] and Cho *et al.* [22] are also based on bilinear pairing operation, therefore, the cost of computation and communication is high.

According to our investigation, [21], [39], [40], [38], [41], [42], and [22] are self-certified public key cryptography-based schemes for vehicular networks. It can be clearly seen that these are not well applicable for time delay-sensitive applications in vehicular networks in terms of overhead or security. Considering the huge advantages brought by the combination of fog computing and vehicular networks, how to achieve efficient and safe authentication of FVNs is a problem worthy of study.

III. Background

In this section, we introduce our FVN system model and make a detailed description of the responsibilities and assumptions of participants. And then, we detail the security objectives and related preliminaries.

A. Network Model and Assumptions

The proposed three-layer architecture of the FVN is shown in Fig. 1. The top layer includes the trusted authority (TA) and the cloud server (CS). The middle layer is a fog layer composed of fog nodes that deployed according to geographic region needs. The bottom layer is a vehicle layer composed of many intelligent vehicles. The following describes each entity and its corresponding responsibilities.

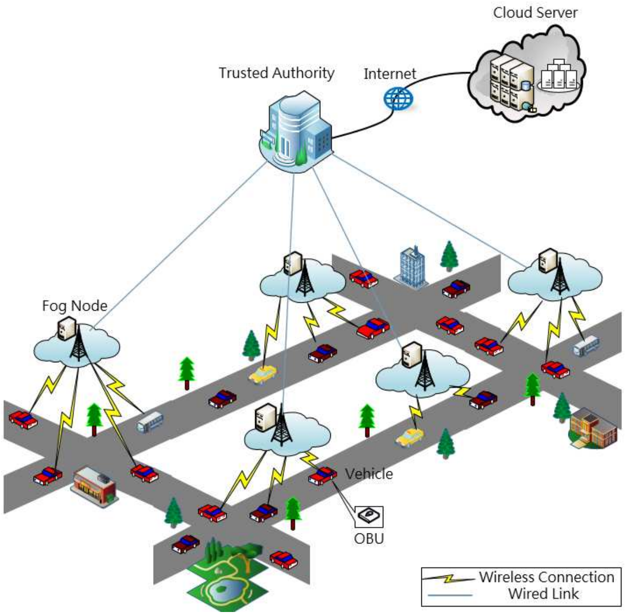


Fig. 1. The three-layer architecture of the fog-based vehicular network.

1) ***TA***: It has sufficient computing and communication capabilities, and is responsible for the initialization of the entire system and the registration of vehicles and fog nodes. When malicious vehicles appear, the TA is the only entity that can track their true identities. Like many

existing schemes, we assume that the TA is completely trustworthy and will not compromise [43].

2) ***Fog node***: The fog node includes a local data storage server and network communication facilities. It is distributed according to the actual regional demand and is geographically closer to the vehicle than the cloud [44]. Fog nodes can process the data uploaded by vehicles and perform regional decision-making, and fog nodes are managed by the TA. Moreover, we assume that fog nodes execute the designated protocols honestly and will not disclose any internal information [33], [45].
3) ***Vehicle***: The vehicle is equipped with an anti-tamper on-board unit (OBU), which supports wireless communication and faithfully performs the set encryption and decryption operations. The vehicle communicates with the fog node via wireless network technology, such as via DSRC, LTE-V2X, or 5G-V2X. We assume that vehicles that participate the vehicular communications are equipped with OBUs, and the secret cryptographic materials stored in OBUs cannot be accessed by anyone. Moreover, most of the vehicles are honest [46].
4) ***CS***: It has huge computing and storage capacity as well as abundant resources. It is responsible for providing remote services for vehicles [47].

Here, we add the CS in Fig. 1 to ensure the integrity of the system, even though the CS does not participate in the encryption process. For example, vehicles will generate and collect a large amount of information, among which data that requires real-time decision-making (e.g., emergency braking, route adjustment) will be processed in time on the vehicle side, and some data (e.g., road condition information) will be uploaded to fog nodes for corresponding data analysis, and some of the remaining data will be uploaded to the CS for data backup (e.g., traffic accident video for accountability) [48].

B. Security Objectives

The security objectives that our protocol aims to achieve are as follows.

1) ***Mutual authentication***: The fog node and the vehicle should verify each other identities and received messages to ensure the reliability of the participants and the integrity of the messages.
2) ***Identity privacy preserving***: In order to realize the privacy protection of the vehicle's real identity, it should be ensured that the real identity of the vehicle is confidential, and no attacker can decipher it through the messages sent by this vehicle.
3) ***Traceability***: If necessary, e.g., when a malicious vehicle sends false information to destroy vehicular communications, the trusted authority can trace the malicious vehicle's real identity for imposing corresponding punishment.
4) ***Session key agreement***: The fog node and vehicle can negotiate a private and secure session key for encrypting subsequent communication.
5) ***Un-linkability***: The passive attacker cannot successfully link the messages sent by the same vehicle through the message content.
6) ***Forward security***: Even if attackers have cracked the current session key, it is still computationally impossible to forge valid signatures for past time periods [49].
7) ***Resistance to ordinary attacks***: In fog-based vehicular networks, attackers can mount passive and active attacks via insure public channels. For launching passive attacks, an attacker merely keeps eavesdropping on the wireless communication and tries to decipher the message; for launching active attacks, an attacker can impersonate another entity and may replay the message. Therefore, the protocol aims to resist the following common attacks.
   - Replay attack: The adversary records the communication session and replays the whole session or part of it at some later point in time [50].
   - Impersonation attack: The adversary successfully assumes the identity of one of the legitimate parties in a protocol.
   - Offline password guessing attack: By analyzing authentication messages offline, the adversary guesses the user's password.

C. Elliptic Curve Cryptography

Elliptic curve cryptography (ECC) is introduced by Victor Miller [51] and Neal Koblitz [52], and has been widely used in various security fields. Let $\mathbb{F}_p$ be a finite field, where p is a prime number. An elliptic curve E over $\mathbb{F}_p$ is the curve defined by: $y^2 = x^3 + ax + b \bmod p$, where $a, b \in \mathbb{F}_p$. P is the generator point of E with a prime order of q. O denotes infinity and $P \neq O$. The elliptic curve group $\mathbb{G}$ has the following properties.

- Additive: Let P and Q be two points of group $\mathbb{G}$. If P and Q are distinct, then we can get a third point $R = P + Q$, where R is intersection of E and the straight line through P and Q. Otherwise, if $P = Q$, then $R = 2P$. If $P = -Q$, then $P + Q = O$.
- Scalar point multiplication: Let $P \in \mathbb{G}$ and $x \in Z_q^*$, the scalar multiplication of E is defined as: $xP = P + P + \cdots + P$.

D. Mathematical Assumptions

The security foundation of our LBVP against a probabilistic polynomial time (PPT) adversary relies on the intractability of the mathematical assumptions as defined in Definitions 1 and 2, i.e., discrete logarithm (DL) assumption and computational Diffie-Hellman (CDH) assumption.

- ***Attack Game on Discrete Logarithm***: Let $\mathbb{G}$ be a elliptic curve group of prime order q generated by $P \in \mathbb{G}$. The challenger $\mathcal{C}$ computes $X = xP$ using the randomly selected element $x \in \mathbb{Z}_q^*$. And $\mathcal{C}$ gives $X \in \mathbb{G}$ to the adversary $\mathcal{A}$. Then $\mathcal{A}$ outputs some $x^{'} \in \mathbb{Z}_q^*$. $\mathcal{A}$'s advantage in solving the DL problem for $\mathbb{G}$, denoted $A_{DL}[\mathbb{A}, \mathbb{G}]$, is defined as the probability that $x^{'} = x$.

Definition 1 (***Discrete Logarithm Assumption***): We say that the DL assumption holds for $\mathbb{G}$ if for all probabilistic polynomial adversaries $\mathcal{A}$ the quantity $A_{DL}[\mathbb{A}, \mathbb{G}]$ is negligible [53].

- ***Attack Game on Computational Diffie-Hellman***: Let $\mathbb{G}$ be a elliptic curve group of prime order q generated by $P \in \mathbb{G}$. The challenger $\mathcal{C}$ computes $X = xP$, $Y = yP$ and $Z = xyP$ using the randomly selected element $x \in \mathbb{Z}_q^*$ and $y \in \mathbb{Z}_q^*$. And $\mathcal{C}$ gives the pair $(X \in \mathbb{G}, Y \in \mathbb{G})$ to the adversary $\mathcal{A}$. Then $\mathcal{A}$ outputs some $Z^{'} \in \mathbb{G}$. $\mathcal{A}$'s advantage in solving the CDH problem for $\mathbb{G}$, denoted $A_{CDH}[\mathbb{A}, \mathbb{G}]$, is defined as the probability that $Z^{'} = Z$.
Definition 2 (***Computational Diffie-Hellman Assumption***): We say that the CDH assumption holds for $\mathbb{G}$ if for all probabilistic polynomial adversaries $\mathcal{A}$ the quantity $A_{CDH}[\mathbb{A}, \mathbb{G}]$ is negligible [53].

IV. LBVP

The proposed lightweight batch verification protocol (L-BVP) for the FVN consists of six phases. The first phase is system setup which is performed by the TA. And then vehicles and fog nodes submit registration applications to the TA and complete the registration in the second and third phases, respectively. Only when the vehicle user successfully passes the fourth login phase, the vehicle will enter the fifth phase of mutual authentication with the fog node. Finally, we provide a user-friendly password change phase that users can perform at any time. Note that, the first three phases only need to perform once. Moreover, in order to show the process of our LBVP more clearly, we show the interactions between these three entities in Fig. 2. And Table II lists the notations.

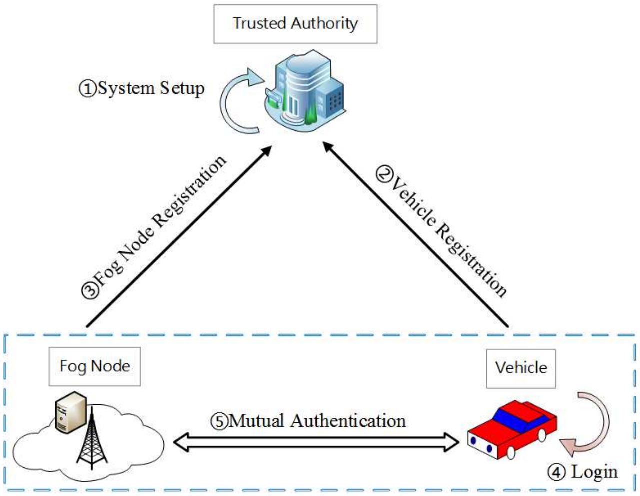


Fig. 2. Framework of the proposed protocol.

Our protocol achieves the following points: 1) it does not need the TA to participate in the authentication phase, therefore, the work burden of the TA is reduced and the overall process of authentication is simplified; 2) the vehicle can update its pseudonym by itself, without complicated interactive steps with the TA; 3) it supports batch verification, which can be performed by fog nodes, to realize rapid message verification; 4) it provides a password change phase that supports users to complete password modification on the vehicle terminal anytime and anywhere.

TABLE II
NOTATIONS

Notations	Definitions
s	Private key of the TA
P_{pub}	Public key of the TA
UID_i	The real identity of user
V_i	The $i-th$ vehicle
ID_i	The real identity of V_i
PID_i	The pseudonym of V_i
PW_i	The password of V_i
F_j	The $j-th$ fog node
ID_{Fj}	The real identity of F_j
T_{i1}, T_{i2}, T_{j1}	The latest timestamp
p, q	Two large prime numbers
E	An elliptic curve
G	An additive group with the order q
P	A generator of the group G
h_i (i=0,...,6)	Collision-free one-way hash function

A. System Setup

First, the TA generates the system parameters. Let F_p be the finite field, and p is a large prime number that denotes the size of the finite field. Then the TA generates an elliptic curve E which defined by $y^2 = x^3 + ax + b \ mod \ p$, and $(a, b) \in F_p$. G is an additive group with the order q and generator P, and G consists of all points on the E. O denotes infinity and $P \neq O$. The TA selects the following one-way hash functions: $h_0 : \{0,1\}^* \rightarrow Z_q^*$, $h_1 : \{0,1\}^* \rightarrow \{0,1\}^l$, $h_2 : \{0,1\}^* \times G \rightarrow Z_q^*$, $h_3 : G \rightarrow Z_q^*$, $h_4 : G \times G \times \{0,1\}^* \rightarrow \{0,1\}^l$, $h_5 : \{0,1\}^* \times \{0,1\}^* \times G \rightarrow Z_q^*$, and $h_6 : G \times \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^l$, where l represents the limited length of bit string. Besides, the TA sets the randomly selected number $s \in Z_q^*$ as its private key, and computes its public key as $P_{pub} = sP$. Finally, the TA publishes the public system parameters $\{P_{pub}, h_0, h_1, h_2, h_3, h_4, h_5, h_6, P, E\}$.

B. Vehicle registration

Each vehicle needs to register with the TA before leaving the factory. After executing the following steps, the vehicle will get a private key. Only by using the private key generated by the TA can the vehicle generate valid pseudo identities to protect its privacy. Following are the details about the interactions between the vehicle V_i and the TA.

1) The vehicle user chooses a password $PW_i \in Z_q^*$ and encrypts the selected login password PW_i using his/her identity UID_i by computing $EPW_i = h_0(UID_i \| PW_i)$. Next, $\{ID_i, EPW_i\}$ will be sent to the TA. Here, the vehicle sends encrypted login passwords EPW_i to the TA is for improving the security of the protocol. Specifically, because the TA does not know the real login password of the vehicle, even if the TA is damaged or launches internal attacks, the login password of the vehicle is still

confidential. Moreover, the attacker cannot obtain the real login password stored in the OBU through side-channel attacks.

2) Upon receiving the registration request submitted by V_i, the TA first checks whether V_i is already registered or in the blacklist, if so, the TA rejects the registration request. Otherwise, the TA continues to perform the following operations.
3) The TA selects a random number $r_i \in Z_q^*$, and then computes $A_i = r_i P$, $B_i = A_i \oplus h_1(ID_i)$, and $C_i = h_2(EPW_i \| A_i)$. Then, the TA generates private key sk_i for V_i by computing $sk_i = r_i + h_3(A_i) \cdot s \ mod \ q$.
4) The TA returns $\{B_i, C_i, sk_i\}$ to V_i.
5) At last, V_i stores $\{B_i, C_i, sk_i, P\}$ into its OBU secretly.

C. Fog node registration

The fog node registers with the TA via a secure channel, such as through TLS protocol [54]. After executing the following steps, the fog node will get a private key for signing its messages. The following is our description of the interactions at this phase.

1) The fog node F_j sends its identity ID_{Fj} to the TA.
2) The TA checks whether the identity ID_{Fj} of the fog node is valid. If not, the TA rejects the registration request. Otherwise, the TA chooses a random number $d_j \in Z_q^*$, and computes $D_{Fj} = d_j P$. Then the TA generates private key sk_{Fj} for F_j by computing $sk_{Fj} = d_j + h_2(ID_{Fj} \| D_{Fj}) \cdot s \ mod \ q$.
3) The TA returns $\{D_{Fj}, sk_{Fj}\}$ to the fog node F_j.
4) At last, the fog node F_j stores $\{sk_{Fj}, P\}$ secretly.

D. Login

The following steps are designed to check the legitimacy of the vehicle user. It should be noted that the time interval for the user to enter the password each time is set in the OBU, and if the number of incorrect password entries exceeds the set threshold, the OBU can stop accepting more input.

1) The vehicle user enters $\{ID_i, PW_i, UID_i\}$ to the OBU of V_i.
2) V_i computes $EPW_i^{'} = h_0(UID_i \| PW_i)$ and $A_i^{'} = B_i \oplus h_1(ID_i)$. If the equation $C_i = h_2(EPW_i^{'} \| A_i^{'})$ holds, this login request will be allowed. Otherwise, the OBU refuses this login request.

E. Mutual Authentication

Due to the high mobility of vehicles, the connection between vehicles and fog nodes is frequently disconnected and established, therefore, it is very important to realize the rapid authentication between them.

1) First, V_i selects a random nonce $x_i \in Z_q^*$ to compute $X_i = x_i P$. Then, V_i generates pseudo-identity PID_i by computing $PID_i = ID_i \oplus h_4(x_i P_{pub} \| A_i \| T_{i1})$. Here, T_{i1} is the latest timestamp. Afterwards, V_i computes $\alpha_i = h_5(PID_i \| T_{i1} \| X_i)$. Finally, V_i signs the message by computing $\sigma_{i1} = sk_i + \alpha_i \cdot x_i \ mod \ q$.
2) V_i sends the message $M_{vi1} = \{X_i, PID_i, A_i, T_{i1}, \sigma_{i1}\}$ to the nearby fog node.
3) Upon receiving the message M_{vi1} from V_i, F_j first checks the validity of timestamp T_{i1} of M_{vi1}. If T_{i1} has expired, F_j drops the message and does not need to perform subsequent operations. Conversely, if T_{i1} is valid, F_j computes $\alpha_i = h_5(PID_i \| T_{i1} \| X_i)$ to verify if the equation (1) holds.

$$\sigma_{i1} \cdot P = A_i + h_3(A_i) \cdot P_{pub} + \alpha_i \cdot X_i \tag{1}$$

If yes, F_j selects a random nonce $y_{Fj} \in Z_q^*$ to compute $Y_{Fj} = y_{Fj} P$. Then, F_j generates the session secret key shared with V_i, where $sek_{ij} = h_6(y_{Fj} X_i \| PID_i \| ID_{Fj})$. F_j computes $\beta_{Fj} = h_5(sek_{ij} \| T_{j1} \| Y_{Fj})$. Finally, F_j signs its response message by computing $\sigma_{Fj2} = sk_{Fj} + \beta_{Fj} \cdot y_{Fj} \ mod \ q$.
4) F_j returns $M_{fj2} = \{Y_{Fj}, ID_{Fj}, D_{Fj}, T_{j1}, \sigma_{Fj2}\}$ to V_i.
5) When V_i gets the message M_{fj2}, the first thing it does is check the validity of the T_{j1}. Similarly, only when the time stamp is valid will the subsequent steps be performed. Otherwise, V_i drops this message. V_i calculates $\beta_{Fj} = h_5(sek_{ij} \| T_{j1} \| Y_{Fj})$ for verifying whether the equation (2) holds, so as to authenticate the messages M_{fj2} sent form F_j.

$$\sigma_{Fj2} \cdot P = D_{Fj} + h_2(ID_{Fj} \| D_{Fj}) \cdot P_{pub} + \beta_{Fj} \cdot Y_{Fj} \tag{2}$$

If the message passes the authentication successfully, V_i calculates the private session key sek_{ij} shared with F_j, where $sek_{ij} = h_6(x_i Y_{Fj} \| PID_i \| ID_{Fj})$. At this point, V_i and F_j complete mutual authentication and establish private session key sek_{ij}.

Remark 1: Since our LBVP supports batch verification, next we will explain in detail how the fog nodes batch verifies multiple messages from different vehicles.

Batch Verification at Fog Node: When the fog node receive multiple messages $M_{vi1} = \{X_i, PID_i, A_i, T_{i1}, \sigma_{i1}\}$, where $i = 1, 2, 3...n$, the following steps will be executed by the fog node. At first, the fog node checks the timestamp T_{i1} in each message, and rejects the message whose timestamp has expired. In order to ensure the non-repudiation of the batch verification signature, we introduce the small exponential test technique into our batch verification process [55]. In small exponent testing, a vector that used to detect any modification of a batch of signatures composed of small random integers, specifically, the fog node randomly chooses a vector $u = \{u_1, u_2, \ldots, u_n\}$, where t is a small random integer and $u_i \in [1, 2^t]$ [43]. And then the fog node computes $\alpha_i = h_5(PID_i \| T_{i1} \| X_i)$ respectively, where $i = 1, ..., n$. Finally, the fog node verifies whether the following equation (3) holds. If yes, the fog node accepts this message tuple.

$$
\begin{aligned}
&(\Sigma_{i=1}^{n} u_i \cdot \sigma_{i1}) \cdot P \\
=&(\Sigma_{i=1}^{n} u_i \cdot (sk_i + \alpha_i \cdot x_i)) \cdot P \\
=&(\Sigma_{i=1}^{n} u_i \cdot sk_i) \cdot P + (\Sigma_{i=1}^{n} u_i \cdot \alpha_i \cdot x_i) \cdot P \\
=&(\Sigma_{i=1}^{n} u_i \cdot (r_i + h_3(A_i) \cdot s)) \cdot P + (\Sigma_{i=1}^{n} u_i \cdot \alpha_i \cdot x_i) \cdot P \\
=&\Sigma_{i=1}^{n} (u_i \cdot A_i) + (\Sigma_{i=1}^{n} u_i \cdot h_3(A_i)) \cdot P_{pub} \\
&+ \Sigma_{i=1}^{n} (u_i \cdot \alpha_i \cdot X_i)
\end{aligned} \tag{3}
$$

Note that, when invalid signatures appear, instead of verifying signatures one by one or discarding the whole signatures, we can adopt binary search technology as described in our previous work [56].

F. Password Change

The vehicle user can change the password by performing the following operations at the vehicle terminal.

1) The user inputs UID_i, ID_i, and PW_i as well as the new password PW_i^* into the OBU of V_i.
2) V_i computes $EPW_i^{'} = h_0(UID_i \| PW_i)$ and $A_i^{'} = B_i \oplus h_1(ID_i)$ to authenticate user's identity. Only when the equation $C_i = h_2(EPW_i^{'} \| A_i^{'})$ holds, V_i calculates $EPW_i^* = h_0(UID_i \| PW_i^*)$ and $C_i^* = h_2(EPW_i^* \| A_i)$ for replacing the password PW_i with PW_i^*. At last, V_i stores $\{B_i, C_i^*, EPW_i^*\}$ secretly.

V. Security Analysis

In this section, we first show the security model, related security definitions, and detailed security analysis successively.

A. Security Model

The security model of our LBVP is defined by a series of games played between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. The participator are a vehicle V_i and a fog node F_j. And let $\prod_{\Lambda}^{k}$ denote the kth instance of $\Lambda \in \{V_i, F_j\}$. $\mathcal{A}$ can execute the following queries to $\mathcal{C}$.

- $Extract\ Vehicle\ (ID_i)$: When $\mathcal{A}$ invokes this query with V_i's identity ID_i, $\mathcal{C}$ executes a key generation algorithm, and stores sk_i to the list L_{VK}.
- $Extract\ Fog\ Node\ (ID_{Fj})$: When $\mathcal{A}$ invokes this query with F_j's identity ID_{Fj}, $\mathcal{C}$ executes a key generation algorithm, and stores sk_{Fj} to the list L_{FK}.
- $Send\ (\prod_{\Lambda}^{k}, m)$: This query models active attacks. If $\mathcal{A}$, using message m, invokes this query, then $\mathcal{C}$ executes the protocol and sends the result to $\mathcal{A}$.
- $Execute\ (V_i, F_j)$: It simulates that $\mathcal{A}$ passive eavesdropping the protocol. The output of this query is the messages that are exchanged by V_i and F_j during the actual execution of our LBVP.
- $Reveal\ (\prod_{\Lambda}^{k})$: When $\mathcal{A}$ invokes this query, if $\prod_{\Lambda}^{k}$ is accepted, then $\mathcal{C}$ returns $\mathcal{A}$ with the session key. Otherwise, $\mathcal{C}$ returns $\perp$, which means no answer is output.
- $Corrupt\ Vehicle\ (V_i, \pi)$: It allows $\mathcal{A}$ to get the following information of V_i.
  - ∗ $\pi = 1$: The password PW_i is obtained by $\mathcal{A}$ via this query.
  - ∗ $\pi = 2$: $\mathcal{A}$ obtains long-term private key of V_i via this query.
- $Corrupt\ Fog\ Node(F_j)$: When $\mathcal{A}$ starts this query with ID_{Fj}, $\mathcal{C}$ returns the long-term private key of F_j to $\mathcal{A}$.
- $Test\ (\prod_{\Lambda}^{k})$: When $\mathcal{A}$ starts this query with instance V_i (the instance V_i needs to satisfy the freshness requirement as defined in Definition 4), $\mathcal{C}$ flips a unbiased coin b, where $b \in \{0, 1\}$. If $b = 1$, $\mathcal{C}$ returns the session key involved in $\prod_{\Lambda}^{k}$ to $\mathcal{A}$; otherwise, $\mathcal{C}$ selects a random number with a length equal to the session key and returns it to $\mathcal{A}$.

Definition 3 (***Partnership***): If the oracle instances V_i and F_j mutually authenticate each other and establish the same session key, and both have the same session and partner identifiers, V_i and F_j are partners [20].

Definition 4 (***Freshness***): The instance V_i is fresh, provided that V_i state is accepted, and V_i meets the following points.

∗ It has not been queried by $Reveal$ query.
∗ Its *partner* (as defined in Definition 3) has not been queried by $Reveal$ query.
∗ Either $Corrupt(V_i, 1)$ or $Corrupt(V_i, 2)$ is not queried by $\mathcal{A}$.
∗ V_i's partner F_j has not been corrupted.

Definition 5 (***Semantic Security***): $\mathcal{A}$ outputs the judgment b involved in the $Test{-}Oracle$, after executing the above queries. $\mathcal{A}$'s ability that success defeat our LBVP is defined as the probability of guessing the b accurately which involved in the $Test{-}Oracle$. That is, the winning advantage of $\mathcal{A}$ is defined as: $Adv_{Lbvp}^{Auth} = |Pr[b = b'] - 1/2| = |Pr[Succ(A)] - 1/2|$. Here, $Pr[Succ(A)]|$ denotes $\mathcal{A}$'s success probability in winning the game $Game(\prod_{\Lambda}^{k}, \mathcal{A})$. If any polynomial adversary cannot win the game with the probability Adv_{Lbvp}^{Auth}, then we say our LBVP is semantically secure.

B. Formal Security Proof

In this section, we will prove that our LBVP satisfies semantic security by proving the Theorem 1.

Theorem 1: No polynomial adversary can win the game with the probability Adv_{Lbvp}^{Auth}, where Adv_{Lbvp}^{Auth} is ignorable:

$$
\begin{aligned}
Adv_{Lbvp}^{Auth}(\mathcal{A}) \leq & \frac{\sum_{i=0}^{6} q_{hi}^2}{2q} + \frac{(q_s + q_e)^2}{2q} \\
& + \frac{q_s}{q} + \frac{q_s}{|D|} + \sum_{i=0}^{6} q_{hi} \cdot Adv_{CDH}(\mathcal{A}) \cdot (t + q_s).
\end{aligned} \tag{4}
$$

During the polynomial time t, $\mathcal{A}$ can perform up to q_{hi} hash-queries (i=0,1,2...6), q_s Send-queries, and q_e Execute-queries. l denotes the length of hash values. D denotes a uniformly distributed password dictionary of length $|D|$.

Proof: Five successive games $G_i (0 \leq i \leq 4)$ are conducted to prove that our LBVP is proven secure. E_i denotes the corresponding event to the G_i, which $\mathcal{C}$ rightly guesses the bit b that existed in the $Test$ query. Moreover, Δ_i represents the difference between $Pr[E_i]$ and $Pr[E_{i-1}]$.

Game G_0: To break the protocol, $\mathcal{A}$ can send various oracle queries to $\mathcal{C}$ and $\mathcal{C}$ responses as follow methods.

- $Extract\ Query$ can be divided into several sub oracles as below according to the specific type of the request.
    1) $Extract\ Vehicle\ (ID_i)$: When $\mathcal{C}$ receives this query, $\mathcal{C}$ checks whether a record (ID_i, sk_i) has appeared. If yes, $\mathcal{C}$ returns $\mathcal{A}$ with ID_i. If not, $\mathcal{C}$ selects $r_i \in Z_q^*$, and computes $A_i = r_iP$ and $sk_i = r_i + h_3(A_i) \cdot s\ mod\ q$. $\mathcal{C}$ stores (ID_i, sk_i) and $(ID_i, h_3(A_i))$ into L_{VK} and L_{h_3}, respectively. Then $\mathcal{C}$ sends ID_i to $\mathcal{A}$.
    2) $Extract\ Fog\ Node(ID_{Fj})$: When $\mathcal{C}$ receives this query, $\mathcal{C}$ checks whether a record (ID_{Fj}, sk_{Fj}) has appeared. If so, $\mathcal{C}$ returns ID_{Fj} to $\mathcal{A}$. Otherwise, $\mathcal{C}$ chooses a random number $d_j \in Z_q^*$, and computes $D_{Fj} = d_jP$ and $sk_{Fj} = d_j + h_2(ID_{Fj}\|D_{Fj}) \cdot s\ mod\ q$. And $\mathcal{C}$ stores (ID_{Fj}, sk_{Fj}) and $(ID_{Fj}, h_2(ID_{Fj}\|D_{Fj}))$ into L_{FK} and L_{h_2}, respectively. Then $\mathcal{C}$ sends ID_{Fj} to $\mathcal{A}$.
- $Send\ Query$ can be divided into several sub oracles as below according to the specific type of the request.
    1) $Send\ (V_i, start)$: When $\mathcal{C}$ receives this query, $\mathcal{C}$ selects $x_i \in Z_q^*$ and computes $X_i = x_iP$, $PID_i = ID_i \oplus h_4(x_iP_{pub}\|A_i\|T_{i1})$, $\alpha_i = h_5(PID_i\|T_{i1}\|X_i)$ and $\sigma_{i1} = sk_i + \alpha_i \cdot x_i\ mod\ q$. Then the query is answered with $M_{vi1} = \{X_i, PID_i, A_i, T_{i1}, \sigma_{i1}\}$.
    2) $Send\ (F_j, M_{vi1})$: When $\mathcal{C}$ receives this query, $\mathcal{C}$ checks the correctness of α_i. If yes, $\mathcal{C}$ selects $y_{Fj} \in Z_q^*$ and computes $Y_{Fj} = y_{Fj}P$, $sek_{ij} = h_6(y_{Fj}X_i\|PID_i\|ID_{Fj})$, $\beta_{Fj} = h_5(sek_{ij}\|T_{j1}\|Y_{Fj})$, and $\sigma_{Fj2} = sk_{Fj} + \beta_{Fj} \cdot y_{Fj}\ mod\ q$. Then the query is answered with $M_{fj2} = \{Y_{Fj}, ID_{Fj}, D_{Fj}, T_{j1}, \sigma_{Fj2}\}$. If not, $\mathcal{C}$ refuses $\mathcal{A}$'s query and returns $\perp$.
    3) $Send\ (V_i, M_{fj2})$: When $\mathcal{C}$ receives this query, $\mathcal{C}$ checks whether β_{Fj} equals to $h_5(sek_{ij}\|T_{j1}\|Y_{Fj})$. If not, $\mathcal{C}$ terminates the game. If yes, $\mathcal{C}$ computes $sek_{ij} = h_6(x_iY_{Fj}\|PID_i\|ID_{Fj})$.
- $Execute\ (V_i, F_j)$: When $\mathcal{C}$ receives the query $Execute\ (V_i, F_j)$ from $\mathcal{A}$, $\mathcal{C}$ recovers (M_{vi1}, M_{fj2}) from the list and then returns (M_{vi1}, M_{fj2}) to $\mathcal{A}$.
- $Reveal\ (\prod_\Lambda^k)$: When $\mathcal{C}$ gets the query $Reveal\ (\prod_\Lambda^k)$ from $\mathcal{A}$, if $\prod_\Lambda^k$ is accepted, $\mathcal{C}$ returns $\mathcal{A}$ with the session key sek_{ij}. If not, $\mathcal{C}$ outputs $\perp$.
- $Corrupt\ Query$ can be divided into several sub oracles as below according to the specific type of the request.
    1) $Corrupt\ Vehicle\ (V_i, \pi)$: When $\mathcal{C}$ receives this query, if $\pi = 1$, $\mathcal{C}$ answers $\mathcal{A}$'s query with the password PW_i.
    2) $Corrupt\ Vehicle\ (V_i, \pi)$: When $\mathcal{C}$ receives this query, if $\pi = 2$, $\mathcal{C}$ answers $\mathcal{A}$'s query with the secret key sk_i of V_i.
    3) $Corrupt\ Fog\ Node(F_j)$: When $\mathcal{C}$ receives this query, $\mathcal{C}$ answers $\mathcal{A}$'s query with the secret key sk_{Fj} of F_j.
- $Test\ (\prod_\Lambda^k)$: When $\mathcal{C}$ receives the query $Test\ (\prod_\Lambda^k)$, $\mathcal{C}$ throws a fair coin $b \in \{0, 1\}$. If b = 1, $\mathcal{C}$ returns $\mathcal{A}$ with the session key that gotten from $Reveal(\prod_\Lambda^k)$; otherwise, $\mathcal{C}$ selects a random number which length is equal to the session key and returns it to $\mathcal{A}$.

It is obvious that G_0 simulates the real attack, all queries are executed in accordance with the protocol specification, and the advantage for $\mathcal{A}$ to break Game G_0 is the same as that of Definition 5. Thus we have

$$Adv_{Lbvp}^{Auth}(\mathcal{A}) = |Pr[E_0] - 1/2|. \tag{5}$$

Game G_1: G_1 simulates the hash oracles and $\mathcal{C}$ maintains the list L_{hi} $(i = 0, 1, ...6)$. When $\mathcal{A}$ starts this query with the message m_i, $\mathcal{C}$ checks if $(m_i, h(m_i))$ in L_{hi}. If yes, $h(m_i)$ will be returned. If not, $\mathcal{C}$ returns $\mathcal{A}$ with a random selected value $h(m_i)$, and then stores $(m, h(m_i))$ into L_{hi}. Since G_1 is perfectly indistinguishable from G_0, therefore, $P[E_1] = P[E_0]$, and we can get

$$\Delta_1 = |Pr[E_1] - Pr[E_0]| = 0. \tag{6}$$

Game G_2: G_2 simulates all the oracles in G_1. If there is a conflict occurs in transcripts and the hash queries, G_1 will stop. Based on the birthday paradox, we get that the maximum probability is $\frac{\sum_{i=0}^{6} q_{hi}^2}{2q}$ for the collision happened on the hash functions. The maximum probability of a collision between random numbers x_i and y_{Fj} is $\frac{(q_s+q_e)^2}{2q}$. Because the difference between G_1 and G_2 cannot be distinguished by $\mathcal{A}$, we have

$$\Delta_2 = |Pr[E_2] - Pr[E_1]| \leq \frac{\sum_{i=0}^{6} q_{hi}^2}{2q} + \frac{(q_s + q_e)^2}{2q}. \tag{7}$$

Game G_3: G_3 simulates all the oracle in G_2. If $\mathcal{A}$ luckily guesses the right verifiers value without asking the hash oracle, G_2 will be aborted. Because the difference between G_2 and G_3 cannot be distinguished by $\mathcal{A}$, therefore,

$$\Delta_3 = |Pr[E_3] - Pr[E_2]| \leq \frac{q_s}{q}. \tag{8}$$

Game G_4: In G_4, we use Elliptic Curve Diffie-Hellman Problem to simulate the executions. Given a CDH instance, we select $\widetilde{x}$ and $\widetilde{y}$ randomly, and compute $X_i = \widetilde{x}P$ and $Y_{Fj} = \widetilde{y}P$. If $Corrupt\ Vehicle\ (V_i, 2)$ query has been issued, it means $Corrupt\ Vehicle\ (V_i, 1)$ cannot be made. Hence, $\mathcal{A}$ can only test a password in each transcript. Therefore, we can get that:

$$\begin{aligned} \Delta_4 &= |Pr[E_4] - Pr[E_3]| \\ &\leq \frac{q_s}{|D|} + \sum_{i=0}^{6} q_{hi} \cdot Adv_{CDH}(\mathcal{A}) \cdot (t + q_s). \end{aligned} \tag{9}$$

- Based on above information, we can get that:

$$\begin{aligned}
&Adv_{Lbvp}^{Auth}(\mathcal{A}) = |Pr[Succ(A)] - 1/2| \\
&= |Pr[Succ(A)] - 1/2 + (Pr[Succ(A)] - Pr[E_m])| \\
&\le |Pr[E_m] - 1/2 + \Sigma_{i=1}^{m-1}\Delta_i| \\
&\le \frac{\sum_{i=0}^{6} q_{hi}^2}{2q} + \frac{(q_s + q_e)^2}{2q} \\
&+ \frac{q_s}{q} + \frac{q_s}{|D|} + \sum_{i=0}^{6} q_{hi} \cdot Adv_{CDH}(\mathcal{A}) \cdot (t + q_s).
\end{aligned} \tag{10}$$

C. Informal Security Analysis

1) ***Mutual authentication***: In our protocol, the fog node and vehicle will authenticate the received messages M_{vi1} and M_{fj2} respectively. And according to our previous analysis, by verifying σ_{i1} and σ_{Fj2}, the vehicle and the fog node can verify the legitimacy of the other party's identity.
2) ***Identity privacy preserving***: In this protocol, the vehicle generates dynamically updated pseudonym PID_i for each communication, where $PID_i = ID_i \oplus h(x_i P_{pub} \| A_i \| T_{i1})$. Because no attacker can crack the elliptic curve discrete logarithm problem, the real identity of the vehicle is confidential and cannot be revealed.
3) ***Traceability***: Because $x_i P_{pub} = sX_i$, therefore, the TA can retrieve V_i's real identity by computing $ID_i = PID_i \oplus h(sX_i \| A_i \| T_{i1})$ using its private key s.
4) ***Session key agreement***: In our LBVP, after the authentication between the vehicle and the fog node, the private session key sek_{ij} known only by both parties will be negotiated, where $sek_{ij} = h_6(x_i Y_{Fj} \| PID_i \| ID_{Fj}) = h_6(y_{Fj} X_i \| PID_i \| ID_{Fj})$.
5) ***Un-linkability***: Because the messages sent by the vehicle contain the latest timestamp and random number used only once, and the pseudonym will be updated dynamically, the attacker can not link the messages from the same vehicle through the message content.
6) ***Forward security***: The session key equals $H(x_i Y_{Fj} \| PID_i \| ID_{Fj})$ and $H(y_{Fj} X_i \| PID_i \| ID_{Fj})$, where x_i and y_{Fj} are numbers selected randomly each time and PID_i is the the pseudo identity of the vehicle. And because attacks do not have the secret key of vehicles, therefore it is computationally impossible for attacks to forge valid past signatures and the forward security is achieved.
7) ***Resistance to ordinary attacks***: Our LBVP is able to resist the following attacks.
   - ***Resistance impersonation of vehicle***: To impersonate a legitimate vehicle and send a valid authentication message, the attacks need to know the correct sk_i and x_i. Because sk_i is stored in the anti-tamper on-board unit of the vehicle, and x_i is the dynamic updated random number and $X_i = x_i P$ involves DL problem, therefore the attacker cannot launch impersonation attacks successfully.
   - ***Resistance replay attack***: Due to timestamp T_{i1}, T_{j1} and T_{i2} are attached to the messages, participants can find whether a replay has occurred by checking the freshness of the timestamp.
   - ***Resistance offline password guessing attack***: Because the password PW_i of vehicle user is stored after encryption, where $C_i = h_2(EPW_i \| A_i)$ and $EPW_i = h_0(UID_i \| PW_i)$. Additionally, users can change the password PW_i frequently, therefore, the adversary can not guess the password PW_i correctly in polynomial time.

VI. Performance Evaluation

To prove that our LBVP achieves better performance, we compare it with He $et\ al.$'s scheme [19], Zeng $et\ al.$'s scheme [26], Wang $et\ al.$'s scheme [18], Shen $et\ al.$'s scheme [10], Azees $et\ al.$'s scheme [57], Song $et\ al.$'s scheme [6], and Ma $et\ al.$'s scheme [13]. Among them, schemes [19], [26], and [18] are based on self-certified public key cryptography, where schemes [10], [57], [6], and [13] are support mutual authentication between vehicles and fog nodes.

A. Computation Cost Analysis

Table III lists the computation cost in each entity of the schemes [19], [26], [18], [10], [57], [6], and [13], and our protocol. To analyze the detailed computation cost on each entity in schemes [10], [57], [6], and [13] and our protocol that are both proposed for vehicular networks, we use the MIRACL library to obtain cryptographic operations' execution time on the Nvidia Drive PX2 which really used in the vehicle (all Tesla Motors vehicles manufactured from mid-October 2016 include a Drive PX 2), the personal computer (HP with an Intel i7-6700 CPU, 8GB DDR4 RAM, and the Ubuntu 14.04 operation system), and the cloud server (ecs.t6-c1m2.large, 2 vCPU, 4 GiB, Intel(R) Xeon(R) Platinum 8269CY). And the tested execution time on the Nvidia Drive PX2, on the personal computer, and on the cloud server are respectively used as the execution time required by the vehicle, the fog node, and the TA/cloud server. Table IV lists the execution time of the basic cryptographic operations on each entity.

Next, we introduce the analysis about Shen $et\ al.$'s scheme [10] and our LBVP in detail, in the same way, we can get the calculation cost analysis of [19], [26], [18], [57], [6] and [13]. The protocol of Shen $et\ al.$'s scheme [10] involved bilinear pairing operations. In the process of the authentication, it requires user to execute one modular exponentiation, one multiplication, one point addition, one bilinear pairing, and four hash operations, that is to say, the execution time on the vehicle is $T_E + T_{Bsm} + T_{Bp} + T_{Bap} + 4T_h \approx 21.8442$ ms. And in this process, it requires the edge server to execute two modular exponentiations, three multiplication, two point addition, two bilinear pairing, and three hash operations; consequently, the execution time on the edge server is $2T_E + 3T_{Bsm} + 2T_{Bp} + 2T_{Bap} + 3T_h$ $\approx$15.548 ms. Consequently, the total time needed for mutual authentication in [10] is $3T_E + 4T_{Bsm} + 3_{Bp} + 3T_{Bap} + 7T_h \approx 37.39$ ms. In our LBVP, the computation time needed in vehicle terminal

TABLE III
COMPUTATION COST

	Vehicle/User	Fog/Edge Server	TA	Total
[19]	$2T_{Bsm} + 1T_{Bap} + 2T_E + 8T_h$	$T_{Bp} + 2T_{Bsm} + 4T_E + 5T_h$	/	$T_{Bp} + 4T_{Bsm} + 1T_{Bap} + 6T_E + 13T_h$
[26]	$2T_{Bsm}+3T_{Bap}+T_E+6T_h+T_s$	$2T_{Bp} + 2T_E + T_M + 3T_h + T_s$	/	$2T_{Bp}+2T_{Bsm}+3T_{Bap}+3T_E+9T_h+2T_s$
[18]	$5T_{Esm} + T_{Eap} + 6T_h$	$5T_{Esm} + 2T_{Eap} + 4T_h$	/	$10T_{Esm} + 3T_{Eap} + 10T_h$
[10]	$T_E+T_{Bsm}+T_{Bp}+T_{Bap}+4T_h$	$2T_E+3T_{Bsm}+2T_{Bp}+2T_{Bap}+3T_h$	/	$3T_E + 4T_{Bsm} + 3_{Bp} + 3T_{Bap} + 7T_h$
[57]	$T_{Bp} + 10T_{Bsm} + 3T_h$	$T_{Bp} + 6T_{Bsm} + 3T_h$	/	$2T_{Bp} + 16T_{Bsm} + 6T_h$
[6]	$11T_{Esm} + 3T_{Eap} + 6T_h$	$8T_{Esm} + 4T_{Eap} + 4T_h$	$T_{Esm} + T_h$	$20T_{Esm} + 7T_{Eap} + 11T_h$
[13]	$3T_{Esm} + 4T_h$	$4T_{Esm} + 4T_h$	$10T_{Esm}+11T_h$	$17T_{Esm} + 19T_h$
Our	$6T_{Esm} + 2T_{Eap} + 4T_h$	$5T_{Esm} + 2T_{Eap} + 4T_h$	/	$11T_{Esm} + 4T_{Eap} + 8T_h$

/: The entity does not need to participate in the authentication phase.

TABLE IV
EXECUTION TIME OF BASIC OPERATIONS (MS)

Symbol	Description	Format	Time$_1$ (ms)	Time$_2$ (ms)	Time$_3$ (ms)
T_{Bp}	Bilinear pairing operation	$\overline{e}(\overline{S},\overline{T})$, where $\overline{S},\overline{T} \in G_1$	13.89	4.669	5.537
T_{Bsm}	Scale multiplication operation related to the bilinear pairing	$\overline{x} \cdot \overline{P}$, where $\overline{P} \in G_1$, $x \in Z_q^*$	2.2078	0.788	0.9104
T_{Bap}	Point addition operation related to the bilinear pairing	$\overline{S}$ + $\overline{T}$, where $\overline{S}$, $\overline{T} \in G_1$	0.0058	0.002	0.0022
T_M	MapToPoint hash operation related to the bilinear pairing	$H_1 : \{0,1\}^* \rightarrow G_1$	0.3036	0.145	0.1174
T_{Esm}	Scale multiplication operation related to the ECC	$x \cdot P$, where $P \in G$ and $x \in Z_q^*$	0.9224	0.341	0.3622
T_{Eap}	Point addition operation related to the ECC	$S + T$, where $S, T \in G$	0.006	0.002	0.0014
T_h	One-way hash function operation	$h : \{0,1\}^* \rightarrow \{0,1\}^l$	0.0044	0.004	0.0006
T_E	Modular exponentiation operation	g^x mod n	5.723	1.915	2.309
T_s	Symmetric encryption/decryption operation	AES-CBC	0.0074	0.003	0.0026

Time$_1$: The execution time on the vehicle. Time$_2$: The execution time on the fog node. Time$_3$: The execution time on the TA/cloud server.

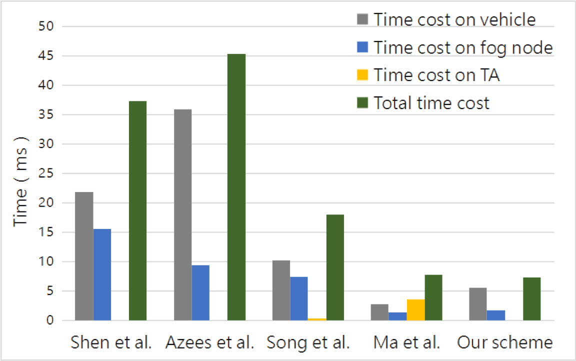


Fig. 3. Comparison of computation cost of schemes that supports vehicle to fog node authentication.

includes six scale multiplication and two point addition along with four one-way hash operations, hence, the execution time on vehicle terminal is $6T_{Esm} + 2T_{Eap} + 4T_h \approx 5.564ms$. The fog node needs to perform five scale multiplication and two point addition along with four one-way hash operations, that is, $5T_{Esm} + 2T_{Eap} + 3T_h \approx 1.725\ ms$. Consequently, the total time needed for mutual authentication in our LBVP is $11T_{Esm} + 4T_{Eap} + 8T_h \approx 7.289\ ms$. From Fig. 3, we can see that compared with schemes [10], [57], [6], and [13], our LBVP achieves lowest total computational cost.

In order to prove the efficiency of batch verification, we compare our LBVP with schemes [10], [57], [6], and [13]. From Fig. 4, we can see that our protocol achieves better batch verification performance. The following is a detailed description of batch verification in each scheme. In Shen *et al.*'s [10] scheme, for batch verifying n messages, the receiver is required to execute $(n + 1)T_{Bp}$ bilinear pairing operations, $2n$ scale multiplication operation related to the bilinear pairing; consequently, the execution time is $(n+1)T_{Bp} + 2nT_{Bsm} \approx$ 6.474n +5.086 ms. In Azees *et al.*'s [57] scheme, they did not consider batch verification, therefore, for batch verifying n messages, the receiver is required to execute n bilinear pairing operations, $5n$ scale multiplication operation related to the bilinear pairing and n general hash function operations;

consequently, the execution time is $nT_{Bp}+5nT_{Bsm}+nT_h \approx 8.613n$ ms. In Song $et\ al.$'s [6] scheme, for batch verifying n messages, the receiver is required to execute $(2n+5)$ scalar multiplication operations, $3n$ point addition operations and $3n$ general hash function operations; consequently, the execution time is $2n+5)T_{Esm}+3nT_{Eap}+3nT_h$ $\approx$0.7n+1.705 ms. In Ma $et\ al.$'s [13] scheme, they did not consider batch verification, therefore, for batch verifying n messages the execution time is $17nT_{Esm}+19nT_h \approx 5.4896n$ ms. That is, $17nT_{Esm}$ scalar multiplication operations related to the ECC and $19nT_h$ general hash function operations.

For batch verifying n messages, in the proposed protocol, the fog node is required to execute $2n$ scalar multiplication operations, $2n$ point addition operations and n general hash function operations; consequently, the execution time is $2nT_{Esm}+2nT_{Eap}+nT_h \approx 0.6494n$ ms. For verifying 100 messages, the computation delay of Shen $et\ al.$'s scheme [10], Azees $et\ al.$'s scheme [57], Song $et\ al.$'s [6], Ma $et\ al.$'s scheme [13], and our LBVP is 652.486 ms, 861.3 ms, 71.705 ms, 548.96 ms, and 64.96 ms, respectively. That is, when verifying 100 messages, our LBVP has improved 90.05%, 92.46%, 9.41%, 88.17% compared with above schemes.

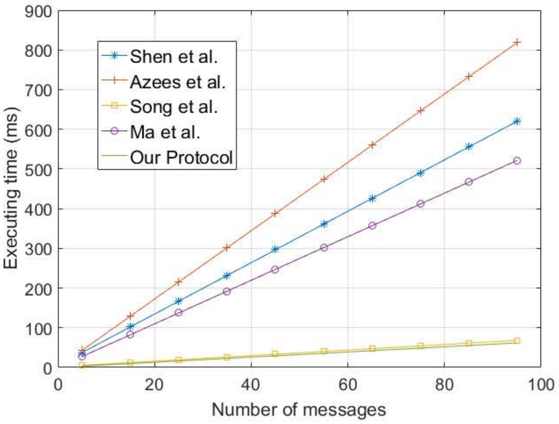


Fig. 4. Delay in the batch verification of multiple messages.

B. Communication Cost Analysis

Because p is 20 bytes and $\overline{p}$ is 64 bytes, the elements in G and G_1 are 20 × 2 = 40 bytes and 64 × 2 = 128 bytes respectively. Moreover, we set the size of timestamp and identity be 4 bytes, and set the size of output of general hash function and symmetric encryption/decryption be 20 bytes, respectively. And we consider the length of messages during authentication phase only. Table V is the communication costs in each protocol during authentication. In the following, we introduce the communication analysis about our LBVP and schemes [19], [26], [18], [10], [57], [6], and [13] in detail.

1) ***Vehicle/User Registration***: In our LBVP, this phase includes two communication rounds, first, vehicle sends $\{ID_i, EPW_i\}$ to the TA for registration and the TA's reply message $\{B_i, C_i, sk_i\}$, where $\langle EPW_i, B_i, C_i, sk_i\rangle$ are the results of one-way hash operation, therefore, the communication cost in this stage is $4+20*4=84\ bytes$. Similarly, the communication costs in [19], [26], [18], [10], [57], [6] , and [13] are 192 bytes, 172 bytes, 104 bytes, 298 bytes, 384 bytes, 40 bytes, and 44 bytes, respectively.

TABLE V
COMMUNICATION COST

	Vehicle/User Registration	Fog Node/Server Registration	Mutual Authentication	Total
[19]	192 bytes	132 bytes	428 bytes	752 bytes
[26]	172 bytes	132 bytes	328 bytes	632 bytes
[18]	104 bytes	64 bytes	244 bytes	412 bytes
[10]	298 bytes	148 bytes	276 bytes	722 bytes
[57]	384 bytes	532 bytes	2148 bytes	3064 bytes
[6]	40 bytes	80 bytes	420 bytes	540 bytes
[13]	44 bytes	44 bytes	600 bytes	688 bytes
Our	84 bytes	64 bytes	248 bytes	396 bytes

2) ***Fog Node/Server Registration***: In our LBVP, this phase includes two rounds, first, fog node sends $\{ID_{Fj}\}$ to the TA for registration than the TA returns $\{D_{Fj}, sk_{Fj}\}$ to fog node, since sk_{Fj} is the result of one-way hash operation, $D_{Fj} \in G$, therefore the communication cost is $4+20+40=64\ bytes$. Similarly, the communication costs in [19], [26], [18], [10], [57], [6], and [13] are 132 bytes, 132 bytes, 64 bytes, 148 bytes, 532 bytes, 80 bytes, and 44 bytes, respectively.

3) ***Mutual Authentication***: In our LBVP, this phase includes two rounds, and the messages are $M_{vi1} = \{X_i, PID_i, A_i, T_{i1}, \sigma_{i1}\}$ and $M_{fj2} = \{Y_{Fj}, ID_{Fj}, D_{Fj}, T_{j1}, \sigma_{Fj2}\}$. Because $\langle X_i, A_i, Y_{Fj}, D_{Fj}\rangle \in G$, and $\langle PID_i, ID_{Fj}, \sigma_{i1}, \sigma_{Fj2}\rangle$ is the results of one-way hash operation, and $\langle T_{i1}, T_{j1}\rangle$ denote the latest timestamp, the communication cost is $40*4+20*4+4*2=248\ bytes$. Similarly, the communication costs during mutual authentication in [19], [26], [18], [10], [57], [6], and [13] are 428 bytes, 328 bytes, 244 bytes, 276 bytes, 2148 bytes, 420 bytes, and 600 bytes, respectively.

Therefore, our LBVP realizes the lowest communication cost in the authentication phase, which decreases by $(428-236)/428 \approx 44.86\%$, $(328-236)/328 \approx 28.05\%$, $(244-236)/244 \approx 3.28\%$, $(722-384)/722 \approx 46.81\%$, $(600-236)/600 \approx 60.67\%$, $(3064-236)/3064 \approx 92.29\%$, and $(540-236)/540 \approx 56.29\%$ respectively, against He $et\ al.$'s scheme [19], Zeng $et\ al.$'s scheme [26], Wang $et\ al.$'s scheme [18], Shen $et\ al.$'s scheme [10], Azees $et\ al.$'s scheme [57], Song $et\ al.$'s scheme [6], and Ma $et\ al.$'s scheme [13].

C. Time Delay Analysis

To compare the network performance, we use Omnet++, Sumo, and Veins [58]. Omnet++ is a component-based C++ simulation library and framework, it is used to construct network simulators that support wired networks and wireless ad hoc networks. Sumo is a continuous road traffic simulation package for handling large road networks. Veins is a middleware connecting the first two modules. The scenario considered is a segment of the real road map around Anhui

University, as shown in Fig. 5. The map comes from OpenStreetMap, which can obtain the attributes of roads in the real world, such as traffic lights, speed limits, etc. The vehicle is on a two-way street and moves in the same direction. Relevant parameters are listed in Table VI.

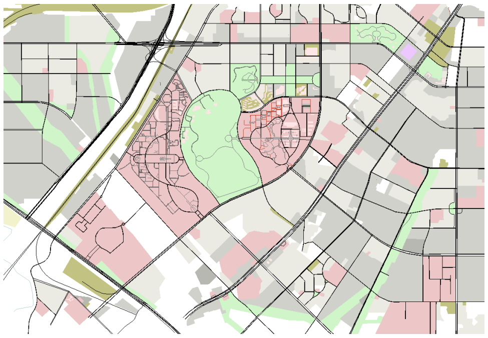


Fig. 5. The simulation map in the experiment.

TABLE VI
SIMULATION PARAMETERS

Parameters	Values
Simulation area	$2500 \times 2500 (m^2)$
Data transmission rate	6 Mbps
Transmission power	50 mW
Sensitivity	-89 dBm
Thermal noise	-110 dBm
Acceleration	10 m/s^2
Simulation time	200 s

Because among schemes [19], [26], [18], [10], [57], [6], and [13], only Azees *et al.*'s scheme [57] and Ma *et al.*'s scheme [13] and our protocol are designed for the fog-based vehicular network, and the communication cost of scheme [57] is obviously higher than ours, therefore, we analysis time delay of Ma *et al.*'s scheme [13] and our protocol. In Ma *et al.*'s scheme [13], the message that vehicle sends to the fog node is $\{AID_{U_i}, T_{U_i}, R_1, \alpha\}$, i.e., the communication overhead is 88 bytes. And the message that fog node returns to the vehicle is $\{R_2, R_3, \hat{R}_3', T_{\mathrm{CS}}, \bar{\gamma}\}$, i.e., the communication overhead is 88 bytes. In the same way, we can get that in our LBVP protocol, the communication overhead of the vehicle to the fog node (V2F) and the fog node to the vehicle (F2V) is 124 bytes and 124 bytes, respectively, as the corresponding message are $M_{vi1} = \{X_i, PID_i, A_i, T_{i1}, \sigma_{i1}\}$ and $M_{fj2} = \{Y_{Fj}, ID_{Fj}, D_{Fj}, T_{j1}, \sigma_{Fj2}\}$.

The comparison result of average wireless transmission delay for the vehicle to the fog node (V2F) communication and fog node to the vehicle (F2V) communication are presented in Fig. 6 and Fig. 7. From Fig. 6 and Fig. 7, we can get three conclusions. The first one is that our protocol achieves a lower average transmission delay of V2F communication. The second one is that the average transmission delay of F2V communication in [13] and our protocol are nearly the same. The third one is that the velocity of the vehicle has a low impact on the transmission delay of V2F and F2V communication.

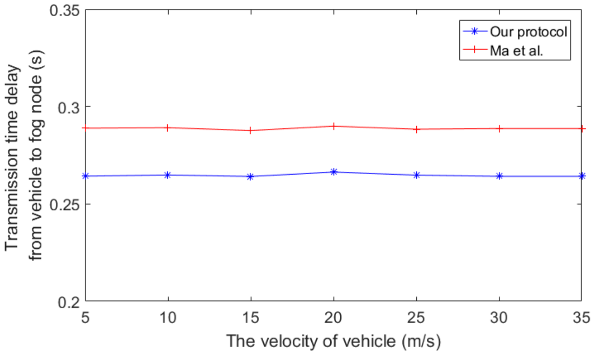


Fig. 6. The comparison of average transmission delay of V2F.

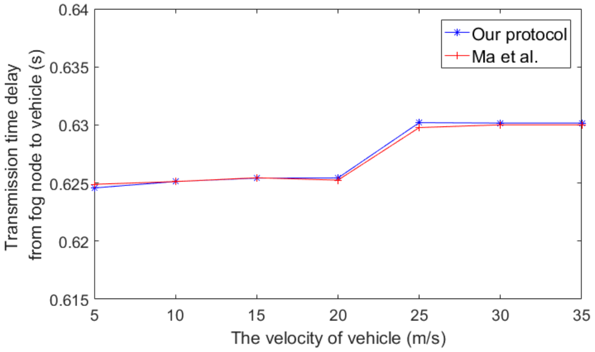


Fig. 7. The comparison of average transmission delay of F2V.

Fig. 8 is the comparison result of total authentication delay under different numbers of vehicles. It should be noted that we define the total authentication delay as the sum of message transmission delay (i.e., V2F communication and F2V communication) and computation delay on each entity in the system model for completing the process of authentication, and when calculating the total authentication delay of Ma *et al.*'s scheme [13], the transmission delay of the wired link, i.e., the wired transmission time from the fog node to the cloud server and from the cloud server to the fog node are not considered. From Fig. 8 we can see that, as expected, because our protocol support batch authentication and can achieve direct authentication between the fog node and the vehicle, our protocol outperforms Ma *et al.*'s scheme [13] in terms of the total authentication delay. And according to the performance requirements formulated by 3GPP [59], the proposed protocol satisfies the latency requirement and is applicable to the vehicle-to-fog computing scenario.

VII. Conclusion

It is of great significance to ensure the security and efficiency of communication between large-scale vehicles and fog nodes in vehicular networks. For the fog-based vehicular network, this paper presented a lightweight batch verification protocol (LBVP), based on self-certified public key cryptography. Because of the use of self-certified public key cryptography, the protocol did not need a trusted third-party authority (i.e., TA) to participate in the authentication process online,

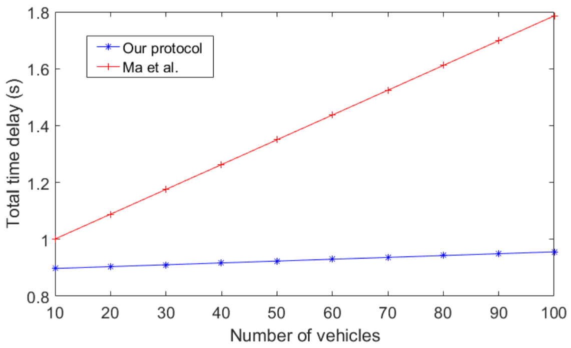


Fig. 8. The comparison of total authentication delay.

and simplified the authentication interaction steps. Moreover, the protocol was based on the elliptic curve cryptosystem, and did not involve bilinear pairing operation, and supported fog node batch verifies multiple messages from vehicles, thus effectively reducing the calculation time delay. The formal security proof under the random oracle model and detailed security analysis shown that the protocol can achieve conditional privacy protection, meet the security requirements of the vehicle network, and resist common types of attacks. The performance comparison with related protocols showed that the protocol achieves lower calculation and communication overhead, and was suitable for fog-based vehicular networks, especially for time delay sensitive application scenarios.

Acknowledgment

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 Excellent Youth Foundation of Anhui Scientific Committee under Grant 2108085J31, 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. The authors are very grateful to the anonymous referees for their detailed comments and suggestions regarding this paper.

References

[1] X. Duan, Y. Liu, and X. Wang, "Sdn enabled 5g-vanet: Adaptive vehicle clustering and beamformed transmission for aggregated traffic," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 120–127, 2017.

[2] C. Campolo, A. Molinaro, A. Iera, and F. Menichella, "5g network slicing for vehicle-to-everything services," *IEEE Wireless Communications*, vol. 24, no. 6, pp. 38–45, 2017.

[3] C. Huang, R. Lu, and K. R. Choo, "Vehicular fog computing: Architecture, use case, and security and forensic challenges," *IEEE Communications Magazine*, vol. 55, no. 11, pp. 105–111, Nov 2017.

[4] 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, February 2019.

[5] J. Kang, R. Yu, X. Huang, and Y. Zhang, "Privacy-preserved pseudonym scheme for fog computing supported internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2627–2637, Aug 2018.

[6] 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.

[7] L.-Y. Yeh, Y.-C. Chen, and J.-L. Huang, "Abacs: an attribute-based access control system for emergency services over vehicular ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 630–643, 2011.

[8] S. Basudan, X. Lin, and K. Sankaranarayanan, "A privacy-preserving vehicular crowdsensing-based road surface condition monitoring system using fog computing," *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 772–782, 2017.

[9] L. Nkenyereye, C. H. Liu, and J. Song, "Towards secure and privacy preserving collision avoidance system in 5g fog based internet of vehicles," *Future Generation Computer Systems*, vol. 95, pp. 488–499, 2019.

[10] 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.

[11] A. Yang, J. Weng, K. Yang, C. Huang, and X. Shen, "Delegating authentication to edge: A decentralized authentication architecture for vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[12] X. Feng, Q. Shi, Q. Xie, and L. Wang, "P2ba: A privacy-preserving protocol with batch authentication against semi-trusted rsus in vehicular ad hoc networks," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3888–3899, 2021.

[13] M. Ma, D. He, H. Wang, N. Kumar, and K.-K. R. Choo, "An efficient and provably secure authenticated key agreement protocol for fog-based vehicular ad-hoc networks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8065–8075, 2019.

[14] M. Cui, D. Han, J. Wang, K.-C. Li, and C.-C. Chang, "Arfv: an efficient shared data auditing scheme supporting revocation for fog-assisted vehicular ad-hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15 815–15 827, 2020.

[15] M. Cui, D. Han, and J. Wang, "An efficient and safe road condition monitoring authentication scheme based on fog computing," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9076–9084, 2019.

[16] J. Cui, L. Wei, H. Zhong, J. Zhang, Y. Xu, and L. Liu, "Edge computing in vanets-an efficient and privacy-preserving cooperative downloading scheme," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1191–1204, 2020.

[17] M. Girault, "Self-certified public keys," *Eurocrypt*, vol. 547, no. 1, pp. 490–497, 1992.

[18] J. Wang, Y. Zhu *et al.*, "Secure two-factor lightweight authentication protocol using self-certified public key cryptography for multi-server 5g networks," *Journal of Network and Computer Applications*, p. 102660, 2020.

[19] D. He, S. Zeadally, N. Kumar, and W. Wu, "Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 2052–2064, Sep. 2016.

[20] D. Abbasinezhad-Mood and M. Nikooghadam, "An anonymous ecc-based self-certified key distribution scheme for the smart grid," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 7996–8004, 2018.

[21] N. M. Rabadi, "Self-certified public key implicit certificate scheme for drivers' anonymity in vehicle-to-vehicle communication networks," in *Proceedings of the IEEE SoutheastCon 2010 (SoutheastCon)*. IEEE, 2010, pp. 375–379.

[22] K. Cho, B.-G. Lee, and D. H. Lee, "Self-certified privacy-preserving scheme for vehicle-to-vehicle communications," in *Computer Science and its Applications*. Springer, 2015, pp. 335–341.

[23] Z. Qin, Y. Li, X. Ye, J. Zhou, M. Cao, and D. Chen, "Ecas: an efficient and conditional privacy preserving collision warning system in fog-based vehicular ad hoc networks," *CCF Transactions on Networking*, vol. 3, no. 3, pp. 205–217, 2020.

[24] I. Ali, Y. Chen, N. Ullah, M. Afzal, and W. He, "Bilinear pairing-based hybrid signcryption for secure heterogeneous vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, 2021.

[25] L. Wei, J. Cui, Y. Xu, J. Cheng, and H. Zhong, "Secure and lightweight conditional privacy-preserving authentication for securing traffic emergency messages in vanets," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1681–1695, 2020.

[26] X. Zeng, G. Xu, X. Zheng, Y. Xiang, and W. Zhou, "E-aua: An efficient anonymous user authentication protocol for mobile iot," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1506–1519, 2018.

[27] F. Wu, L. Xu, X. Li, S. Kumari, M. Karuppiah, and M. S. Obaidat, "A lightweight and provably secure key agreement system for a smart grid

with elliptic curve cryptography," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2830–2838, 2018.

[28] M. J. Hossain, C. Xu, C. Li, S. H. Mahmud, X. Zhang, and W. Li, "Icas: Two-factor identity-concealed authentication scheme for remote-servers," *Journal of Systems Architecture*, vol. 117, p. 102077, 2021.

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

[30] Q. Kong, R. Lu, F. Yin, and S. Cui, "Privacy-preserving continuous data collection for predictive maintenance in vehicular fog-cloud," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2020.

[31] C. Tang, X. Wei, C. Zhu, Y. Wang, and W. Jia, "Mobile vehicles as fog nodes for latency optimization in smart cities," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2020.

[32] D.-N. Vu, N.-N. Dao, W. Na, and S. Cho, "Dynamic resource orchestration for service capability maximization in fog-enabled connected vehicle networks," *IEEE Transactions on Cloud Computing*, 2020.

[33] 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*. IEEE, 2008, pp. 1229–1237.

[34] D. Huang, S. Misra, M. Verma, and G. Xue, "Pacp: An efficient pseudonymous authentication-based conditional privacy protocol for vanets," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 736–746, 2011.

[35] A. Haripriya and K. Kulothungan, "Ecc based self-certified key management scheme for mutual authentication in internet of things," in *2016 International Conference on Emerging Technological Trends (ICETT)*. IEEE, 2016, pp. 1–6.

[36] D. Li, H. Chen, C. Zhong, T. Li, and F. Wang, "A new self-certified signature scheme based on ntrus ing for smart mobile communications," *Wireless Personal Communications*, vol. 96, no. 3, pp. 4263–4278, 2017.

[37] H. Li, F. Guo, L. Wang, J. Wang, B. Wang, and C. Wu, "A blockchain-based public auditing protocol with self-certified public keys for cloud data," *Security and Communication Networks*, vol. 2021, 2021.

[38] N. M. Rabadi, "Revised self-certified implicit certificate scheme for anonymous communications in vehicular networks," in *2010 IEEE Vehicular Networking Conference*. IEEE, 2010, pp. 286–292.

[39] N. M. Rabadi, "Implicit certificates support in ieee 1609 security services for wireless access in vehicular environment (wave)," in *The 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2010)*. IEEE, 2010, pp. 531–537.

[40] W. Li, Q. Wen, Q. Su, and Z. Jin, "An efficient and secure mobile payment protocol for restricted connectivity scenarios in vehicular ad hoc network," *Computer Communications*, vol. 35, no. 2, pp. 188–195, 2012.

[41] X. Wang, Z. Huang, Q. Wen, and H. Zhang, "An efficient anonymous batch authenticated and key agreement scheme using self-certified public keys in vanets," in *2013 IEEE International Conference of IEEE Region 10 (TENCON 2013)*. IEEE, 2013, pp. 1–4.

[42] J. Zhang, Y. Cui, and Z. Chen, "Spa: self-certified pkc-based privacy-preserving authentication protocol for vehicular ad hoc networks," *International Journal of Security and Its Applications*, vol. 6, no. 2, pp. 409–414, 2012.

[43] 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. 12, pp. 2681–2691, 2015.

[44] C. H. Tseng, S. Wang, and W. Tsaur, "Hierarchical and dynamic elliptic curve cryptosystem based self-certified public key scheme for medical data protection," *IEEE Transactions on Reliability*, vol. 64, no. 3, pp. 1078–1085, Sep. 2015.

[45] A. J. Kadhim and S. A. H. Seno, "Energy-efficient multicast routing protocol based on sdn and fog computing for vehicular networks," *Ad Hoc Networks*, vol. 84, pp. 68–81, 2019.

[46] J.-L. Huang, L.-Y. Yeh, and H.-Y. Chien, "Abaka: An anonymous batch authenticated and key agreement scheme for value-added services in vehicular ad hoc networks," *IEEE Transactions on vehicular technology*, vol. 60, no. 1, pp. 248–262, 2010.

[47] M. Tao, K. Ota, and M. Dong, "Foud: Integrating fog and cloud for 5g-enabled v2g networks," *IEEE Network*, vol. 31, no. 2, pp. 8–13, March 2017.

[48] W. Zhang, Z. Zhang, and H. Chao, "Cooperative fog computing for dealing with big data in the internet of vehicles: Architecture and hierarchical resource management," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 60–67, Dec 2017.

[49] G. Itkis and L. Reyzin, "Forward-secure signatures with optimal signing and verifying," in *Annual International Cryptology Conference*. Springer, 2001, pp. 332–354.

[50] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. CRC press, 2018.

[51] V. S. Miller, "Use of elliptic curves in cryptography," in *Conference on the theory and application of cryptographic techniques*. Springer, 1985, pp. 417–426.

[52] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of computation*, vol. 48, no. 177, pp. 203–209, 1987.

[53] D. Boneh and V. Shoup, "A graduate course in applied cryptography," 2020.

[54] H. Vasudev and D. Das, "An efficient authentication and secure vehicle-to-vehicle communications in an iov," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. IEEE, 2019, pp. 1–5.

[55] 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.

[56] J. Cui, J. Zhang, H. Zhong, and Y. Xu, "Spacf: A secure privacy-preserving authentication scheme for vanet with cuckoo filter," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10 283–10 295, 2017.

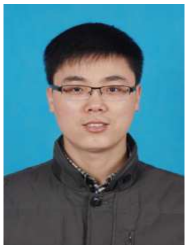
[57] M. Azees, P. Vijayakumar, and L. J. Deboarh, "Eaap: Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2467–2476, 2017.

[58] L. Wei, J. Cui, H. Zhong, Y. Xu, and L. Liu, "Proven secure tree-based authenticated key agreement for securing v2v and v2i communications in vanets," *IEEE Transactions on Mobile Computing*, 2021.

[59] D. Garcia-Roger, E. E. González, D. Martín-Sacristán, and J. F. Monserrat, "V2x support in 3gpp specifications: From 4g to 5g and beyond," *IEEE Access*, vol. 8, pp. 190 946–190 963, 2020.

Xiaoyu Zhang is currently a Ph.D. Student in the School of Computer Science and Technology, Anhui University, Hefei, China. Her research focuses on vehicle ad hoc network.

Hong Zhong was born in Anhui Province, China, in 1965. She received her PhD 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.



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.

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 Department 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 Liu 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).