eCLAS: An Efficient Pairing-Free Certificateless Aggregate Signature for Secure VANET Communication

Yibo Han, Wei Song, Zhangbing Zhou, Hao Wang, and Bo Yuan

***Abstract*—Vehicular ad hoc networks (VANETs) have become an important part of the intelligent transportation system that aims to promote communication among vehicles to ensure vehicular safety and improve the driving experience. In VANETs, signed messages need to be authenticated by roadside units in a very short time. Certificateless aggregate signature (CLAS) scheme has been considered as one of the promising solutions to solve the problem of limited network bandwidth and computing power in VANETs environment. In this article, we propose an efficient pairing-free CLAS, named eCLAS, which is suitable for communication between vehicle to infrastructure. The aggregated signature allows individual signature on different messages from different vehicles to be aggregated into one short signature. In the random oracle model, the security of the scheme is proved by the adaptive selection message attack and the difficulty of computation of discrete logarithm problem on the elliptic curve. Additionally, the security analysis shows that the proposed scheme can meet security requirements in VANETs. Compared with existing schemes, the result demonstrates that our scheme has obvious advantages in signature verification, thereby granting it better applicability.**

***Index Terms*—Aggregate signature, authentication, conditional privacy, elliptic curve cryptography (ECC), vehicular ad hoc networks (VANETs).**

I. Introduction

VEHICULAR ad hoc networks (VANETs) have attracted widespread concern from academia and industry owing to their advantages in road safety and traffic management. The so-called VANET is a type of mobile ad hoc network, which can provide vehicle-to-infrastructure (V2I) and vehicle-to-vehicle communication [1]. The former refers to the information interaction between vehicles and infrastructure, such as the roadside units (RSUs), whereas the latter means the communication between vehicles and neighboring vehicles. A typical VANET structure comprises four parts: trusted authority (TA), application server (AS), RSUs, and on-board units (OBUs) embedded in the vehicles. TA is usually undertaken by an intelligent transportation system department of government, and OBU is tamper-proof equipment that has good wireless communication capabilities [2]. The vehicle sends/receives safety-related messages (such as speed, location, direction, and dangerous road conditions) to/from nearby vehicles and RSUs, so as to achieve the primary target in VANETs. The primary target is improving the safe travel and driving experience of drivers and passengers by sharing data.

In 1999, the United States Federal Communications Commission allocated 75 MHz of spectrum at 5.9 MHz to be used by dedicated short-range communication (DSRC) [3]. DSRC specifies that a vehicle should broadcast basic safety message every 100–300 ms, and the maximum data rate supported by this standard is 27 Mb/s. The mobility support is up to 100 km/h [4]. As vehicles in VANETs communicate through wireless channels, attackers are able to stage various attacks by controlling the communication channels [5]. Therefore, it is essential to ensure the safety-related information is authenticated, undeniable, and unmodified. The digital signature technology has been widely used to authenticate messages, and it is used in this study to achieve the same [6].

Vehicle privacy is also a factor to be considered. In VANETs, safety-related messages include vehicle's traffic trajectory, position, direction, etc. These can be used to infer sensitive private information about the drivers. Therefore, it is important for vehicles to use pseudonyms instead of real identities during communication to achieve privacy protection. Moreover, in the event of traffic collisions or crimes, the TA as an authoritative third party should be able to track the real identity of vehicle through messages. Hence, conditional privacy protection mechanism needs to be designed [7], [8].

In VANETs, there is a situation in which vehicles send signature messages at the same time under the condition of high density traffic. Moreover, considering the limitation of network bandwidth, the transmission of a large number of messages would lead to considerable computational and communication overheads. Aggregate signature is an effective technique for mitigating aforementioned problems, as it is a type of many-to-one mapping that maps multiple signatures from different vehicles

Manuscript received August 26, 2020; revised November 27, 2020, May 7, 2021, and August 9, 2021; accepted September 19, 2021. Date of publication October 19, 2021; date of current version March 24, 2022. *(Corresponding author: Bo Yuan.)*

Yibo Han is with the Nanyang Institute of Big Data Research, Nanyang Institute of Technology, Nanyang 473004, China (e-mail: hanyibo@nyist.edu.cn).

Wei Song is with the School of Computer and Software, Nanyang Institute of Technology, Nanyang 473004, China (e-mail: vivisongysu@163.com).

Zhangbing Zhou is with the School of Information Engineering, China University of Geosciences, Beijing 100083, China (e-mail: zbzhou@cugb.edu.cn).

Hao Wang is with the School of Digital Media and Art Design, Nanyang Institute of Technology, Nanyang 473004, China (e-mail: wanghaoxc@163.com).

Bo Yuan is with the School of Electronics, Computing and Mathematics, University of Derby, DE22 1GB Derby, U.K. (e-mail: yuanbotiankong@126.com).

Digital Object Identifier 10.1109/JSYST.2021.3116029

to a single signature [9]. The validity of the aggregated signature is guaranteed by verifying the validity of each signature involved. This characteristic of aggregate signature can greatly reduce computational and communication overheads, makes it particularly suited to bandwidth-constrained VANETs.

Several works have been done to propose the authentication mechanism in VANETs, which can be divided into based on public key infrastructure (PKI) and based on identity (ID). The PKI-based mechanism needs to manage the certificate pool for the public key of the vehicle by the certification authority (CA), whereas the RSU or vehicle requires additional calculations for verifying the other certificates [10]. ID-based mechanisms are used to mitigate the computation and communication burden, however they are existing the problem of key escrow issues [6], [11]. In this article, a novel pairing-free certificateless aggregate signature (CLAS) scheme, referred to as eCLAS, is proposed. It eliminates complex and time-consuming bilinear pairing.

A. Motivation and Contribution

In VANETS, for every 100–300 ms, hundreds of messages will be sent to RSUs. In V2I communications, the schemes [6], [7], [10], [11] mentioned earlier contain operations, such as bilinear pairings as well as map-to-point hash functions. To reduce the computational cost and time in verification process, this motivates us to design an efficient pairing-free CLAS scheme. In addition, we found that schemes in [6] and [12] cannot resist the type I adversary A_1's and the type II adversary A_2's attacks. To overcome the weakness, we propose a secure and efficient CLAS scheme. There are three main contributions of the proposed eCLAS scheme, which are as follows.

1) We present an efficient authentication mechanism that allows the ASs to quickly verify the feedback given by the RSUs. No longer constrained by time-consuming bilinear pairing, the proposed scheme only uses lightweight cryptography to implement signature and verification functions. This speeds up the performance of V2I communication.
2) We prove the security of the eCLAS scheme with respect to existential unforgeability against Type I and II attacks in the random oracle model. The scheme meets the various security requirements in the VANETs, such as conditional privacy of the vehicles, message integrity and authentication, and resistance to different attacks.
3) A detailed comparison with existing related schemes in terms of security, computational overhead, communication overhead, and storage cost verifies that our scheme shows better performance.

B. Structure of This Article

The rest of this article is organized as follows. A survey of related works is shown in Section II. Some preliminaries and background information are presented in Section III. Section IV describes the proposed eCLAS scheme. An in-depth security analysis is given in Section V. The performance evaluation of the proposed scheme is presented in Section VI. Finally, Section VII concludes this article.

II. Related Work

The integrity, authentication, and nonrepudiation of messages are provided by digital signatures in VANETs. For V2I communication, each RSU may verify too many signed messages in the scenario of high-density traffic, leading to a rather high computational overhead [13]. In 2003, the concept of aggregate signature was first proposed by Boneh *et al.* [14]. Aggregate signature allows individual signatures on different messages from different vehicles to be aggregated into one short signature. Due to the limitation of network bandwidth, the aggregate signature is a very useful technology to weaken the communication cost and computational cost.

At present, some PKI-based signature schemes have been designed, where the CA needs to manage the certificate pool for the public keys of vehicles. And because vehicles and RSUs need to store and verify certificates, the storage cost and calculation overhead of the system are large. For this reason, some researchers proposed identity-based (ID-based) public key encryption schemes. However, in the ID-based scheme of Gentry and Ramzan [15], since the private key of each signer consists of two group elements, it brings secret parameter storage problems to each signer. Yi-ling *et al.* [16] pointed out that the ID-based scheme of Cheng *et al.* [17] cannot resist existential forgery attacks, and proposed a new scheme, but the length of signature increases linearly, resulting in a huge verification cost. Yu *et al.* [18] proposed a new ID-based signature scheme, however, it is vulnerable to parameter substitution attacks. Cui *et al.*'s [19] scheme supports secure and privacy-preserving cooperative downloading, however, the scheme does not specify how to achieve authentication between vehicles and edge computing vehicles.

It can be seen that the aforementioned ID-based authentication schemes are not satisfactory. Moreover, because in some ID-based schemes, the TA stores the private keys of all registered vehicles [20]. Once the system is broken or subjected to internal attacks, the privacy of vehicles will be leaked and the security of the system will be threatened [21]. To solve the key escrow problems in ID-based schemes, Al-Riyami and Paterson [22] first proposed a certificateless public key cryptography (CL-PKC) signature scheme in 2003. On this basis, researchers proposed a large number of certificateless signature (CLS) schemes. Yum and Lee [23] constructed a CLS scheme and claimed that it can resist attacks based on identity and adaptive selection messages. Unfortunately, the scheme [23] could not resist the public key substitution attack. Later, Zhang *et al.* [24] proposed an efficient CLS scheme, which improved the security model of CLS scheme.

For reducing the length of signature and ease the pressure of network bandwidth, the CLAS scheme combines the advantages of the aggregate signature and CL-PKC. In the CLAS scheme, the signature size and verification overhead are greatly reduced. Due to this advantage, the CLAS scheme has been widely used in various scenarios and received extensive attention. In 2007, Gong *et al.* [25] first proposed two CLAS schemes and defined the safety model of the CLAS scheme, however, there are some shortcomings in the security model. Xiong *et al.* [26] provided a provably safe CLAS scheme, and only constant pairing calculation is needed during the verification, but the scheme is not secure and the signature can be forged. In 2015, Horng *et al.* [6] provided a new certificateless signature scheme and an efficient CLAS scheme with conditional privacy-preserving for vehicular sensor networks. However, their scheme was insecure against the malicious-but-passive KGC attack. In [27], Ali *et al.* proposed a blockchain-based certificateless public key signature scheme for V2I communication in VANETs, which is based on bilinear pairing operation. A scheme that uses bilinear pairings and supports signature aggregation verification was proposed by Kumar *et al.* [7]. Similarly, based on bilinear pairing operation, in [28], Mei *et al.* proposed a CLAS scheme with conditional privacy preservation for VANETs. In addition, bilinear pairing operation is also involved in [29] and [30]. However, it is well known that bilinear pairing operation requires expensive computation cost, which is not very suitable for delay-sensitive vehicular networks. Recently, Zhong *et al.* [11] proposed a full-aggregation certificateless signature scheme, unfortunately, their scheme is insecure against a signature-forgery attack by a type II adversary. In [12], Kamil and Ogundoyin proposed an improved CLAS scheme without bilinear pairings for VANETs, however, the scheme could not resist forgery attack.

According to the aforementioned, it can be seen that although the CLAS scheme is very attractive for time-delayed vehicular applications, it is still challenging to design a secure and efficient CLAS-based scheme for VANET communication.

III. Preliminaries and Background

For the design of the proposed eCLAS scheme, we describe some preliminary knowledge, the system model, and the security requirements in this section.

A. Elliptic Curve Cryptosystem and Assumptions

An elliptic curve E with a finite field $Z_p^* = \{1, 2, \ldots, q-1\}$ is defined by the following formula: $y^2 = x^3 + ax + b(\text{mod } p)$, where $a, b \in Z_p^*$, $(4a^3 + 27b^2)(\text{mod} p) \neq 0$. The point and infinity point Θ of the curve form an elliptic curve additive group G_p. The properties of the elliptic curve group are described as follows.

1) *Point addition:* Suppose two random points P_1 and P_2 on E. $P_1 + P_2 = P_3$ if $P_1 \neq P_2$, the line connecting P_1 and P_2 intersects E at $-P_3$, otherwise $P_3 = 2P_1$ if $P_1 = P_2$, the line connecting P_1 and P_2 is the tangent of curve E.
2) *Scalar point multiplication:* Suppose the scalar multiplication or point multiplication on E is given as: $mP = P + P + \cdots + P$ (m times), where $m \in Z_p^*, m > 0$.
3) *Order of a point:* n is the order of a point if p is smallest integer number that make $np = \Theta$ and $n > 0$.

Definition 1: Elliptic Curve Discrete Logarithm Problem (ECDLP): On E, $x \in Z_q^*$ and $Q = xP$, where $P, Q \in G$, G has the prime order q and generator P. The computation of a number x is hard such that $Q = xP$.

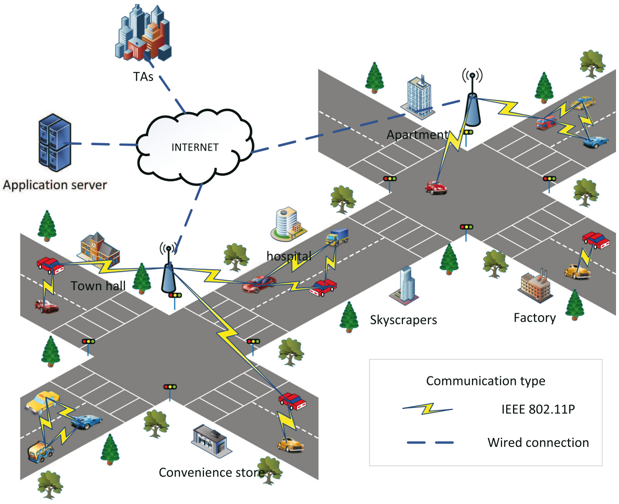

Fig. 1. System model.

Definition 2: Computational Diffie–Hellman Problem: On E, the random points $Q = xP$ and $R = yP$, where $P, Q \in G$, G has the prime order q and generator P. The computation of $xyP \in G$ is hard, where $x, y \in Z_p^*$ are two unknown random numbers.

B. System Model

As shown in Fig. 1, our two-layer system model contains five entities: ASs, RSUs, vehicles equipped with OBUs, and two TAs [i.e., tracing authority (TRA) and key generation center (KGC)]. The upper layer consists of AS, KGC, and TRA, where they could communicate with each other through a secure channel that can be established through the secure socket layer protocol. The lower level is composed of vehicles and RSUs. Vehicles communicate with RSUs through the DSRC protocol. RSUs communicate with ASs and TAs via a secure transport protocol (such as a wired transport layer security protocol) [31].

The details of each entity are as follows.

1) *AS:* It can authenticate its received messages from RSU, and help collect and analyze traffic conditions, so as to predict traffic distribution for optimizing traffic light control. Besides, it could provide vehicles with video conferencing, route recommendation, driving assistance services, etc. It has enough computing and storage capabilities.
2) *KGC:* It is in charge of generating system parameters and assigning the private key to the vehicles and RSUs. It is credible and will not compromise or collude with others. It has enough computing and storage capacity.
3) *TRA:* The TRA first registers the RSU and vehicle having an OBU. To illustrate, TRA is the only entity that can track the true identity of a vehicle. It is assumed that the TRA will never be compromised. It has enough computing and storage capabilities.
4) *Vehicle:* Each vehicle equips an OBU, which is a tamper-proof device, that can prevent the adversary from acquiring data stored in it. The OBUs have limited computing power.

5) *RSU:* As an intermediate entity between OBU, TRA, and KGC, it is a base-station fixed along a roadside and hot spots. It is responsible for aggregating signatures from vehicles and has more computing power than the OBU.

C. Security Requirements

The scheme is supposed to satisfy the following security requirements.

1) *Anonymity:* If a vehicle transmits a message containing its true identity to RSUs or other vehicles in plain text over public channel, there will be serious problems of identity exposure. Therefore, the real identity of the vehicle must remain anonymous to other entities.
2) *Traceability:* Even if the real identity of the vehicle is hidden from other entities in the VANETs, the TRA should be able to get its real identity for tracking the malicious behavior of the vehicle and handle it accordingly.
3) *Unlinkability:* No entity could know that two or more messages are sent from the same vehicle.
4) *Message authentication and integrity:* In the communication process of V2I, RSUs verify the validity of the messages and signatures sent by the vehicle, and ensure it is not modified by other entities.

Resistance to different attacks: The proposed scheme should ensure against the following common attacks.

1) *Replay attack:* An attacker obtains the information transmitted in the middle and repeats them for illegal access to the secret content of the communication process.
2) *Impersonation attack:* The adversary disguises himself as a legal vehicle by modifying the authentication information. If such attack is not dealt with properly, some criminal acts may be carried out by malicious vehicles, resulting in a preplanned traffic accident, etc.
3) *Modification attack:* An attacker has the ability to delete, modify, and change the content of the message for achieving intended target.
4) *Message spoofing attack:* An attacker sends spoof messages, such as false road congestion messages, which interfere with the VANETs and affect the behavior of other vehicles.
5) *Man-in-the-middle attack:* An attacker, respectively, establishes contact with the communication entities and controls the whole conversation, whereas the entities still think that they talk with each other through a private connection.

IV. Proposed eCLAS Scheme

In this section, we will introduce our elliptic curve cryptography (ECC) based CLAS scheme in detail. Our scheme involves m RSUs ($\mathrm{RSU}_1, \mathrm{RSU}_2, \mathrm{RSU}_3, \ldots, \mathrm{RSU}_m$) and n vehicles ($V_1, V_2, V_3, \ldots, V_n$). The solution focuses on achieving secure V2I communication and it includes six algorithms: system initialization, pseudonym identity generation, vehicle key generation, partial private key extract, individual sign, and verification as well as aggregation signature and verification. Fig. 2 shows the network diagram of our scheme. The following content describes each algorithm in detail. Table I provides the notations used in the proposed eCLAS scheme.

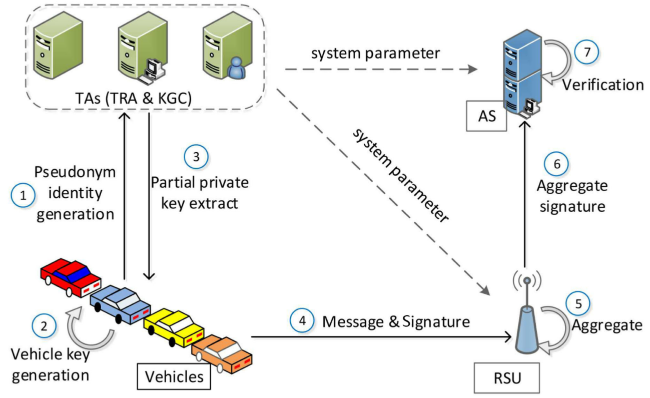

Fig. 2. System model of video multicast in SDVN.

TABLE I
Descriptions of Different Notations

Notations	Descriptions
p, q	Larger prime numbers
Z_p^*	Finite field
V_i	The i^{th} vehicle
RSU_j	The j^{th} road-side-unit
OBU_i	The i^{th} on-board-unit attached to the V_i
E	An elliptic curve: $y^2 = x^3 + ax + b \mod q$
G_p	An addition group
P	The generator of the addition group G_p
a, K_{pub}	KGC's master secret and public keys
b, T_{pub}	TRA's master secret and public keys
vsk_{PID_i}, vpk_{PID_i}	Vehicle V_i's private and public keys
psk_{PID_i}	Vehicle V_i's partial private key
RID_i, PID_i	Vehicle V_i's real and pseudonym identities

A. System Initialization

This algorithm is generate system initialization parameters under the control of TRA and KGC. Detailed steps are as follows.

1) The TRA and KGC use a security parameter λ. The TA TRA and KGC select a group G with prime order q and a generator P on elliptic curve E, as described in preliminaries section.
2) The KGC picks a random number $a \in Z_p^*$ as its master private key for partial private key extract and calculates the corresponding public key $K_{\text{pub}} = aP$, where a is only known by the KGC. TRA also picks $b \in Z_p^*$ randomly as its master private key for vehicle identity tracking and calculates the system public key $T_{\text{pub}} = bP$, where b is only known by the TRA.
3) KGC and TRA select three security hash functions $h_1 : \{0,1\}^* \to Z_p^*$, $h_2 : \{0,1\}^* \to Z_p^*$, and $h_3 : \{0,1\}^* \to Z_p^*$ and publish system parameter $params = \{P, p, q, E, G, h_1, h_2, h_3, K_{\text{pub}}, T_{\text{pub}}\}$.
4) After the system parameter $params$ is released, ASs can get the corresponding system parameters. Besides, any

vehicle V_i sends RID_i to TAs for registration, and the system parameter $params$ is obtained in a secure manner and stored in the OBU of vehicle V_i. At the same time, any RSU is also registered during the initialization phase and the system parameters $params$ are obtained in a safe manner.

B. Pseudonym Identity Generation

In this section, the TRA calculates and generates the pseudonymous identity for the vehicles. Vehicles' messages are sent in a pseudonym way for protecting their real identity information not leaked. At the same time, through the pseudonym identity, conditional privacy can be achieved, because if necessary, the real identity of the vehicle can be revealed by the TRA through the messages. The specific process is as follows.

1) V_i selects a random number $x_i \in Z_p^*$, calculates $\mathrm{PID}_{i,1} = x_i P, R_i = x_i T_{\text{pub}} \oplus \mathrm{RID}_i$, then it sends the message $\{\mathrm{PID}_{i,1}, R_i\}$ to TRA.
2) Upon receiving the message $\{\mathrm{PID}_{i,1}, R_i\}$ from V_i, TRA calculates $\mathrm{RID}_i = R_i \oplus b\mathrm{PID}_{i,1}$ and verifies the validity of this identity, that is, to ensure this vehicle is legal and has not been registered. If the verification fails, TRA will discard this message directly. Otherwise, the TRA calculates $\mathrm{PID}_{i,2} = \mathrm{RID}_i \oplus h_1(b\mathrm{PID}_{i,1} \parallel \Delta T_i)$ and sends the pseudonym identity $\mathrm{PID}_i = \{\mathrm{PID}_{i,1}, \mathrm{PID}_{i,2}, \Delta T_i\}$ to the KGC, where ΔT_i is the pseudonym's validity period.

Here, we adopt the preload method to quantify the upper and lower bounds of pseudonym identity change interval to maintain a satisfactory level of privacy. The vehicle uses the aforementioned algorithm to load a pseudoidentity pool using a short expiration time. When the network is available at some time and freely updatable, the pseudonym pool will be reassembled through the safe channel of the vehicle and TAs.

C. Vehicle Key Generation

The vehicle V_i selects a random number $vsk_{\mathrm{PID}_i} \in Z_p^*$ as its private key, then calculates the corresponding public key $vpk_{\mathrm{PID}_i} = vsk_{\mathrm{PID}_i} P$. Next, the vehicle V_i publishes its public key vpk_{PID_i}, that is, other entities in the VANETs can also obtain the public key.

D. Partial Private Key Extract

1) After KGC receives the pseudonym identity $\mathrm{PID}_i = \{\mathrm{PID}_{i,1}, \mathrm{PID}_{i,2}, \Delta T_i\}$ of vehicle V_i from TRA, it calculates $Q_{\mathrm{ID}_i} = h_2(\mathrm{PID}_i \parallel vpk_{\mathrm{PID}_i}), W_i = Q_{\mathrm{ID}_i} K_{\text{pub}}$, and thus obtains a partial private key $psk_{\mathrm{PID}_i} = aQ_{\mathrm{ID}_i}$.
2) The KGC sends the vehicle's pseudonym identity and partial private key $\{\mathrm{PID}_i, psk_{\mathrm{PID}_i}\}$ to the vehicle V_i in a secure manner and saves it in its corresponding OBU. Therefore, the signature private key of vehicle V_i can be expressed as $\{vsk_{\mathrm{PID}_i}, psk_{\mathrm{PID}_i}\}$.

E. Individual Sign and Verification

To ensure the authentication and the integrity of message, each message sent by the vehicle must be signed and the receiver should verify the message when it receives the message. Vehicle V_i in the range of RSU_j selects a pseudonym identity PID_i and the current timestamp T_i. The traffic-related message M_i is signed with the signature private key $vsk_{\mathrm{PID}_i}, psk_{\mathrm{PID}_i}$, and the vehicle sends the signed message every 100–300 ms. Detailed steps are given as follows.

1) Vehicle V_i selects $r_i \in Z_p^*$ randomly, and calculates $U_i = r_i P$.
2) Vehicle V_i calculates $h_i = h_3(M_i \parallel \mathrm{PID}_i \parallel vpk_{\mathrm{PID}_i} \parallel U_i \parallel T_i)$ and $S_i = psk_{\mathrm{PID}_i} + vsk_{\mathrm{PID}_i} h_i$, then $\sigma_i = (U_i, S_i)$ is the certificateless signature corresponding to message M_i.
3) Vehicle V_i sends message $\{\mathrm{PID}_i, vpk_{\mathrm{PID}_i}, M_i, T_i, \sigma_i\}$ to the nearby RSU_j or ASs for verification.
4) When RSU_j or an AS receives message $\{\mathrm{PID}_i, vpk_{\mathrm{PID}_i}, M_i, T_i, \sigma_i\}$ from vehicle V_i, if ΔT_i of pseudonym identity is legal, and T_i is within the valid time interval, then it calculates $Q_{\mathrm{ID}_i} = h_2(\mathrm{PID}_i \parallel vpk_{\mathrm{PID}_i})$ and $h_i = h_3\{M_i \parallel \mathrm{PID}_i \parallel vpk_{\mathrm{PID}_i} \parallel U_i \parallel T_i\}$. Next, it checks whether $S_i P = W_i + vpk_{\mathrm{PID}_i} h_i$ is established. If it is established, the verification is passed and the certificate is received. Otherwise, the signature is discarded and the vehicle V_i certification fails. Because $S_i = psk_{\mathrm{PID}_i} + vsk_{\mathrm{PID}_i} h_i, psk_{\mathrm{PID}_i} = aQ_{\mathrm{ID}_i}$,and $Q_{ID_i} = h_2(PID_i \parallel vpk_{\mathrm{PID}_i})$. From this, we can get

$$\begin{aligned} S_i P &= (psk_{\mathrm{PID}_i} + vsk_{\mathrm{PID}_i} h_i) P \\ &= psk_{\mathrm{PID}_i} P + vsk_{\mathrm{PID}_i} h_i P \\ &= a h_2(\mathrm{PID}_i \parallel vpk_{\mathrm{PID}_i}) P + vpk_{\mathrm{PID}_i} h_i \\ &= W_i + vpk_{\mathrm{PID}_i} h_i. \end{aligned}$$

F. Aggregate Signature and Verification

Compressing the signature length through the idea of aggregate signature can effectively relieve the pressure of the transmission process when the network bandwidth is limited. The specific aggregate signature and verification process are described as follows.

1) When RSU_j receives multiple messages $\{\mathrm{PID}_i, vpk_{\mathrm{PID}_i}, M_i, T_i, \sigma_i\}$, $i \in (1, 2, 3, \ldots, n)$ from different vehicles $(V_1, V_2, V_3, \ldots, V_n)$ with certificateless message-signature pair $\{(M_1, \sigma_1), (M_2, \sigma_2), (M_3, \sigma_3), \ldots, (M_n, \sigma_n)\}$, RSU_j first checks the validity of ΔT_i in pseudonym identity in each message, and if one T_i in a message is incorrect, the aggregate signature should be recalculated again; otherwise it performs the following steps.
2) RSU_j acts as an aggregate signature generator, aggregating multiple certificateless signatures into one short signature. That is, RSU_j calculates $S = \Sigma_{i=1}^n S_i$ and outputs $\sigma = (U_1, U_2, U_3, \ldots, U_n, S)$ as a CLAS to facilitate later verification of aggregate signature.
3) Once an AS receives a CLAS $\sigma = (U_1, U_2, U_3 , \ldots, U_n, S)$ from RSU_j signed by n vehicles $(V_1, V_2, V_3, \cdots, V_n)$ with pseudoidentities $(\mathrm{PID}_1, \mathrm{PID}_2, \mathrm{PID}_3, \ldots, \mathrm{PID}_n)$ and the corresponding public

keys $(vpk_{\text{PID}_1}, vpk_{\text{PID}_2}, vpk_{\text{PID}_3}, \ldots, vpk_{\text{PID}_n})$ on messages $\{M_1, M_2, M_3, \ldots, M_n\}$, AS calculates $Q_{\text{ID}_i} = h_2(\text{PID}_i \parallel vpk_{\text{PID}_i})$ and $h_i = h_3(M_i \parallel \text{PID}_i \parallel vpk_{\text{PID}_i} \parallel U_i \parallel T_i)$, $i \in \{1, 2, 3, \ldots, n\}$ separately.

4) AS checks whether $SP = \Sigma_{i=1}^{n} W_i + \Sigma_{i=1}^{n} vpk_{\text{PID}_i} h_i$ is established. If it is true, then the aggregate signature verification is passed and these messages are accepted; otherwise, perform the following steps. Because $S = \Sigma_{i=1}^{n} S_i$, $S_i = psk_{\text{PID}_i} + vsk_{\text{PID}_i} h_i$, $psk_{\text{PID}_i} = aQ_{\text{ID}_i}$, $Q_{ID_i} = h_2(\text{PID}_i \parallel vpk_{\text{PID}_i})$, $K_{\text{pub}} = aP$, $W_i = Q_{\text{ID}_i} K_{\text{pub}}$, and $vpk_{\text{PID}_i} = vsk_{\text{PID}_i} P$, we can get that

$$\begin{aligned} SP &= \Sigma_{i=1}^{n} S_i P \\ &= \Sigma_{i=1}^{n} (psk_{\text{PID}_i} + vsk_{\text{PID}_i} h_i) P \\ &= \Sigma_{i=1}^{n} a h_2(\text{PID}_i, vpk_{\text{PID}_i}) P + \Sigma_{i=1}^{n} vpk_{\text{PID}_i} h_i \\ &= \Sigma_{i=1}^{n} W_i + \Sigma_{i=1}^{n} vpk_{\text{PID}_i} h_i. \end{aligned}$$

5) When invalid signatures appear, instead of verifying signatures one by one or discarding the whole signatures, we can verify the aggregate signatures by using binary search technology highlighted in Algorithm 1 [32].

Specifically, when the aggregate signature verification fails, the AS informs the RSU_j via the secure channel that some signatures are invalid in its aggregated signature. Then, the received signatures are sorted in a certain order. The intermediate point is first found and the signatures are divided into two parts. The signatures of these two parts are reaggregated and sent to the AS for verification. If either of the two portions fails in the batch verification phase once again, we will do identical operations on the invalid batch repeatedly. Unless a batch of signatures just includes one signature, the binary search will stop. This process can effectively avoid the problem that once the aggregation verification fails, all the signatures are invalid.

6) *Batch verification:* In this section, we introduce the small exponent test technology to guarantee the nonrepudiation of signatures, that is the RSU_j generates the vector $(v_1, v_2, v_3, \ldots, v_n)$, $v_i \in [1, 2^t]$, where t is a very small integer that does not generate computational overhead. Next, RSU_j checks whether $(\Sigma_{i=1}^{n} v_i S_i)P = \Sigma_{i=1}^{n}(v_i W_i) + \Sigma_{i=1}^{n}(v_i vpk_{\text{PID}_i} h_i)$ is established. If it is established, the verification is passed and the certificate is received, and RSU_j performs the following verification operation:

$$\begin{aligned} &(\Sigma_{i=1}^{n} v_i S_i)P \\ &= (\Sigma_{i=1}^{n} v_i (psk_{\text{PID}_i} + vsk_{\text{PID}_i} h_i))P \\ &= \Sigma_{i=1}^{n} v_i psk_{\text{PID}_i} P + \Sigma_{i=1}^{n} v_i vsk_{\text{PID}_i} h_i P \\ &= \Sigma_{i=1}^{n} v_i a h_2(\text{PID}_i, vpk_{\text{PID}_i}) P + \Sigma_{i=1}^{n} v_i vpk_{\text{PID}_i} h_i \\ &= \Sigma_{i=1}^{n}(v_i W_i) + \Sigma_{i=1}^{n}(v_i vpk_{\text{PID}_i} h_i) \end{aligned}$$

Algorithm 1: Invalid Signature Search $(L, L1, low, high)$.

```
Require: $L = \{M_1, M_2, M_3, \ldots, M_n\}$.
List $L1 = Null$ is used to store invalid messages
if $AggregateSignatureVerification(L, low, high)$ then
  return 1
else
  if $low == high$ then
    $L1.append(L[low])$
    return 1
  else
    $mid = (low + high)/2$
    $InvalidSignatureSearch(L, L1, low, mid)$
    $InvalidSignatureSearch(L, L1, mid + 1, high)$
    return 1
  end if
end if
```

V. Security Analysis

In this section, we first prove that our certificateless signature scheme is secure against adaptively selected message attacks under the random oracle model, and we further prove that our eCLAS scheme is secure. At last, we make detailed security analysis to demonstrate that our scheme can meet the privacy requirements in the VANETs.

Forking Lemma [33]: Let A be a probabilistic polynomial time Turing machine whose input only consists of public data. We denote, respectively, by Q and R the number of queries that A can ask to the random oracle and the number of queries that A can ask to the signer. Assume that, within a time bound T, A produces, with probability $\varepsilon \geq 10(R+1)(R+Q)/2^k$, a valid signature $(m, \sigma_1, h, \sigma_2)$. If the triples (σ_1, h, σ_2) can be simulated without knowing the secret key, with an indistinguishable distribution probability, then there is another machine that has control over the machine obtained from A replacing interaction with the signer by simulation and produces two valid signatures $(m, \sigma_1, h, \sigma_2)$ and $(m, \sigma_1, h', \sigma_2')$ such that $h \neq h'$ in expected time $T' \geq 120686QT/\varepsilon$.

A. Formal Security Analysis

Theorem 1: The proposed eCLAS scheme is unforgeable for adaptively selecting messages under the random oracle model. According to the forking lemma [33], we can derive this theorem from Lemma 1.

Lemma 1: Under the random oracle model, a probabilistic polynomial-time adversary A_1 forges a certificateless signature in an attack modeled by the forking lemma after making q_{h_2} times h_2 queries, q_{h_3} times h_3 queries, q_{crev} times create vehicle queries, q_{ppk} times partial private key queries, q_{seck} times secret key oracle queries, and q_{sign} times sign oracle queries. If the adversary A_1 has the advantage of forging an valid signature in polynomial-time, there is a challenger C_1 that can solve ECDLP in time T expected to be less than $120686QT/\varepsilon$, if $\varepsilon \geq 10(q_{\text{sign}} + 1)(q_{h_2} + q_{h_3} + q_{\text{ppk}} + q_{\text{crev}} + q_{\text{seck}} + q_{\text{sign}})/q$.

Proof: Assuming given $P, Q = xP$, where P and Q are two points on elliptic curve E, forger A_1 can forge message $\{\text{PID}_i, vpk_{\text{PID}_i}, M_i, T_i, \sigma_i\}$. We have built a game between A_1

and a challenger C_1, and C_1 has the ability to run A_1 with a nonnegligible probability as a subroutine to solve ECDLP.

Setup: The master key a is randomly selected by challenger C_1. And then C_1 calculates the corresponding public key $K_{\text{pub}} = aP$. Next, C_1 sends the system parameter $params = \{P, p, q, E, G, h_1, h_2, h_3, K_{\text{pub}}, T_{\text{pub}}\}$ to A_1.

h*₂ *queries: When A_1 uses the parameter $(\text{PID}_i, vpk_{\text{PID}_i})$ for h_2 query, C_1 checks whether the tuple $(\text{PID}_i, vpk_{\text{PID}_i}, \tau_{h_2})$ already exists in the hash list L_{h_2}. If it is, C_1 sends $\tau_{h_2} = h_2(\text{PID}_i, vpk_{\text{PID}_i})$ to A_1. Otherwise C_1 selects $\tau_{h_2} \in Z_q^*$ randomly and adds the elements $(\text{PID}_i, vpk_{\text{PID}_i}, \tau_{h_2})$ to the hash list L_{h_2}, then C_1 sends $\tau_{h_2} = h_2(\text{PID}_i, vpk_{\text{PID}_i})$ to A_1.

h*₃ *queries: When A_1 uses the parameter $(M_i, \text{PID}_i, vpk_{\text{PID}_i}, U_i, T_i)$ for a h_3 query, C_1 determines whether the tuple $(M_i, \text{PID}_i, vpk_{\text{PID}_i}, U_i, T_i, \tau_{h_3})$ already exists in the hash list L_{h_3}. If it is, C_1 sends $\tau_{h_3} = h_3(M_i,\ \text{PID}_i,\ vpk_{\text{PID}_i},\ U_i,\ T_i)$ to A_1. Otherwise C_1 selects a random number $\tau_{h_3} \in Z_q^*$ and adds the tuple $(M_i, \text{PID}_i, vpk_{\text{PID}_i}, U_i, T_i, \tau_{h_3})$ to the hash list L_{h_3}, then C_1 sends $\tau_{h_3} = h_3(M_i, \text{PID}_i, vpk_{\text{PID}_i}, U_i, T_i)$ to A_1.

Partial private key queries: When A_1 performs a partial private key query on the pseudonym identity, C_1 calculates $Q_{\text{ID}_i} = h_2(\text{PID}_i, vpk_{\text{PID}_i})$, $W_i = Q_{\text{ID}_i} \cdot K_{\text{pub}}$ and determines if tuple $(\text{PID}_i, vpk_{\text{PID}_i}, \tau_{h_2})$ already exists in hash list L_{h_2}. If C_1 cannot find the corresponding tuple, it outputs fails and halts because it cannot answer the query consistently. Otherwise, C_1 calculates $psk_{\text{PID}_i} = aQ_{\text{ID}_i}$ and sends psk_{PID_i} to A_1.

Create vehicle queries: Assume this request is a request for pseudonym identity.

1) If list L contains $(\text{PID}_i, vpk_{\text{PID}_i}, vsk_{\text{PID}_i})$, C_1 checks if $vpk_{\text{PID}_i} = \perp$ holds. If $vpk_{\text{PID}_i} = \perp$, C_1 sends vpk_{PID_i} to A_1. Otherwise, C_1 picks a random number as its private key $vsk_{\text{PID}_i} \in Z_q^*$ and computes $vpk_{\text{PID}_i} = vsk_{\text{PID}_i}P$, meanwhile, C_1 sends vpk_{PID_i} to A_1 and updates $(vpk_{\text{PID}_i}, vsk_{\text{PID}_i})$ to list L.
2) If list L does not involve $(\text{PID}_i, vpk_{\text{PID}_i}, vsk_{\text{PID}_i})$, C_1 sets $vpk_{\text{PID}_i} = \perp$ and picks a random number as its private key $vsk_{\text{PID}_i} \in Z_q^*$, and calculates $vpk_{\text{PID}_i} = vsk_{\text{PID}_i}P$, next, C_1 sends vpk_{PID_i} to A_1 and updates $(\text{PID}_i, vpk_{\text{PID}_i}, vsk_{\text{PID}_i})$ to list L.

Secret key queries: Assume this request is a request for secret key.

1) If list L involves $(\text{PID}_i, vpk_{\text{PID}_i}, vsk_{\text{PID}_i})$, C_1 checks whether $vsk_{\text{PID}_i} = \perp$ is true. If $vsk_{\text{PID}_i} = \perp$, C_1 sends vsk_{PID_i} to A_1. Otherwise, C_1 performs a create vehicle queries to generate $vpk_{\text{PID}_i} = vsk_{\text{PID}_i}P$. After that, C_1 sends vsk_{PID_i} to A_1 and updates $(vpk_{\text{PID}_i}, vsk_{\text{PID}_i})$ to list L.
2) If list L does not involve $(\text{PID}_i, vpk_{\text{PID}_i}, vsk_{\text{PID}_i})$, C_1 performs a create vehicle query and sends vsk_{PID_i} to A_1. After that, C_1 sends vsk_{PID_i} to A_1 and updates $(\text{PID}_i, vpk_{\text{PID}_i}, vsk_{\text{PID}_i})$ to list L.

Sign query: When A_1 performs this query on message M_i, C_1 first checks if element $(\text{PID}_i, vpk_{\text{PID}_i}, \tau_{h_2})$ is in list L_{h_2}. If no, C_1 gets τ_{h_2} from the tuple and selects two random numbers r_i, h_i. In addition, C_1 calculates $U_i = r_iP$ and $S_i = h_iP$, sends $\sigma_i = (U_i, S_i)$ to A_1, and adds $(\text{PID}_i, vpk_{\text{PID}_i}, M_i, T_i, \sigma_i, \tau_{h_3})$ to list L_{h_3}.

On the basis of the forking lemma [33], C_1 has the ability to get two different valid signatures $\sigma_i = (U_i, S_i)$ and $\sigma_i' = (U_i', S_i')$ in polynomial time through A_1, where $S_i = psk_{\text{PID}_i} + vsk_{\text{PID}_i}h_i$ and $S_i' = psk_{\text{PID}_i} + vsk_{\text{PID}_i}h_i'$. Because

$$\begin{aligned}\frac{h_i'S_i - h_iS_i'}{h_i' - h_i} &= \frac{h_i'(psk_{\text{PID}_i} + vsk_{\text{PID}_i}h_i) - h_i(psk_{\text{PID}_i} + vsk_{\text{PID}_i}h_i')}{h_i' - h_i} \\ &= \frac{h_i'psk_{\text{PID}_i} + h_i'vsk_{\text{PID}_i}h_i - h_ipsk_{\text{PID}_i} - h_ivsk_{\text{PID}_i}h_i'}{h_i' - h_i} \\ &= psk_{\text{PID}_i}\end{aligned}$$

for $\varepsilon \geq 10(q_{\text{sign}} + 1)(q_{h_2} + q_{h_3} + q_{\text{ppk}} + q_{\text{crev}} + q_{\text{seck}} + q_{\text{sign}})/q$, C_1 can solve ECDLP within a desired time less than $120686QT/\varepsilon$. However, this is in contradiction with the difficulty of ECDLP. Therefore, our certificateless signature scheme can resist forgery attacks.

Theorem 2: If the proposed certificateless signature algorithm is secure against adaptive chosen message attacks, then the proposed CLAS algorithm is equally secure against existential forgery in the chosen aggregation model.

Proof: Suppose there is a forger A_2 that can destroy the CLAS algorithm. We show how a challenger C_2 exploits the ability of A_2 to solve the ECDLP. C_2 and A_2 interactively to perform the following simulation.

Setup: The master key $a \in Z_q^*$ is randomly selected by challenger C_2. And it calculates the corresponding public key $K_{\text{pub}} = aP$ as well as executes the oracle simulation. When A_2 queries the entire game, C_2 maintains a list $L = (\text{PID}_i, psk_{\text{PID}_i}, vpk_{\text{PID}_i}, vsk_{\text{PID}_i})$, and responds A_2's oracle query in the following way.

h*₂ *queries: When the pseudonym identity PID_i is submitted to the oracle h_2, C_2 throws a coin $c_i \in \{0, 1\}$ to generate the probability, that is, the probability ε if generate 0, and the probability $1 - \varepsilon$ if generate 1. And C_2 randomly picks $w_{1i} \in Z_q^*$.

1) If $c_i = 0$, Q_{ID_i} is defined as $Q_{\text{ID}_i} = h_2(\text{PID}_i, vpk_{\text{PID}_i})$, and set $psk_{\text{PID}_i} = w_{1i}Q_{\text{ID}_i}$.
2) If $c_i = 1$, C_2 calculates $psk_{\text{PID}_i} = w_{1i}Q_{\text{ID}_i}$.

In both cases, C_2 inserts an element $(\text{PID}_i, w_{1i}, c_i, Q_{\text{ID}_i})$ in list $L_{h_2} = (\text{PID}_i, w_{1i}, c_i, Q_{\text{ID}_i})$ to track how it responds to the query.

When A_2 outputs the pseudoidentities of n vehicles from the set $L_{\text{PID}}^* = \{\text{PID}_1^*,\ \text{PID}_2^*,\ \text{PID}_3^*,\ \cdots,\ \text{PID}_n^*\}$, the public keys $L_{vpk}^* = \{vpk_{\text{PID}_1}^*, vpk_{\text{PID}_2}^*, vpk_{\text{PID}_3}^*, \ldots, vpk_{\text{PID}_n}^*\}$ corresponds to each anonymous identity, n messages $L_M^* = \{M_1^*,\ M_2^*,\ M_3^*,\ \cdots,\ M_n^*\}$ as well as a CLAS $\sigma^* = \{U_1^*,\ U_2^*,\ U_3^*,\ \cdots,\ U_n^*,\ S^*\}$. C_2 finds the corresponding n tuple $(\text{PID}_i, w_{i1}, c_i, Q_{\text{m}_i})$ from L_{h_2} with $c_k = 1$ and $c_j = 1$, and $i = 1, 2, 3, \ldots, n$. Here, $(\text{PID}_k^*, pvk_{\text{PID}_k}^*, M_k^*)$ has not been sent to the sign oracle. Otherwise, C_2 fails and halts. If C_2 successes, that means $Q_{\text{ID}_k} = h_2(\text{PID}_k, vpk_{\text{PID}_k}), psk_{\text{PID}_i} = w_{1i}Q_{\text{ID}_i}, j = 1, 2, 3, \ldots, n, j \neq k.$, and the aggregate signature satisfy $SP = \Sigma_{i=1}^{n}W_i + \Sigma_{i=1}^{n}vpk_{\text{PID}_i}h_i$.

After that, C_2 finds the corresponding tuples $(M_i^*, \text{PID}_i^*, vpk_{\text{PID}_i}^*, U_i^*, w_{2i}^*)$ and $(\text{PID}_i^*, psk_{\text{PID}_i}^*, vpk_{\text{PID}_i}^*, vsk_{\text{PID}_i}^*)$ from the lists L_{h_3} and L_{h_2}, respectively. Next, C_2 sets $S_i^* = w_{1i}^* a, S_i^* P = w_{1i}^* K_{\text{pub}} = W_i^*$. Finally, C_2 builds $S^{'*} = S^* - \Sigma_{i=1, i\neq k}^n S_i^*$ corresponding to $U_i^* = r_i^* P$, so $S^{'*} = psk_{\text{PID}_i^*}^* - \Sigma_{i=1}^n w_{2i}^* r_i^*$. C_2 picks $h_k^* \in Z_q^*$ randomly and computes $U_i^{'*} = (h_k^*)^{-1} \Sigma_{i=1}^n w_{2i}^* U_i^*$. After that, C_2 computes the hash function value $h_k^* = h_3(M_k^*, \text{PID}_k^*, vpk_{\text{PID}_k^*}^*, U_k^*)$, and $vpk_{\text{PID}_k^*}^{'*} = \Sigma_{i=1}^n vpk_{\text{PID}_i^*}^*$. If the tuple $h_3(M_k^*, \text{PID}_k^*, vpk_{\text{PID}_k^*}^*, U_k^*)$ is already in the list L_{h_3}, C_2 will try again until it does not happen. So, $(U^{'*}, S^{'*})$ is a valid certificateless signature for the message M_k^*. According to the following verification equation, a certificateless signature scheme can be forged:

$$
\begin{aligned}
& W_k^* + \Sigma_{i=1}^n vpk_{\text{PID}_k^*}^{'*} h_k^* \\
& = psk_{\text{PID}_k^*}^* P + \Sigma_{i=1}^n vpk_{\text{PID}_k^*}^{'*} h_k^* \\
& = psk_{\text{PID}_k^*}^* P + \Sigma_{i=1}^n vsk_{\text{PID}_k^*}^* h_k^* P = S^{'*} P.
\end{aligned}
$$

However, this is a contradiction in the difficulty of ECDLP assumptions. Therefore, our scheme can resist such attacks.

B. Informal Security Analysis

1) *Anonymity:* In the proposed eCLAS scheme, all vehicles use pseudonyms to participate in the communication in the VANETs. Because the pseudonym sent by the vehicle is in the form of $\text{PID}_i = \{\text{PID}_{i,1}, \text{PID}_{i,2}, \Delta T_i\}$. In order to obtain the true identity of the target vehicle, the attacker calculates $\text{RID}_i = R_i \oplus b\text{PID}_{i,1}$ or $\text{RID}_i = \text{PID}_{i,2} \oplus h_1(b\text{PID}_{i,1} \parallel \Delta T_i)$. However, because these two methods both involve ECDLP that attackers cannot solve, attackers cannot get the real identity of vehicles through the messages, thus the scheme realizes the protection of users' anonymity.
2) *Traceability:* All entities other than TRA do not have a master private key b. Given a pseudonym identity $\text{PID}_i = \{\text{PID}_{i,1}, \text{PID}_{i,2}, \Delta T_i\}$ on the signature message, the TRA can use the primary and private keys b to calculate $\text{RID}_i = \text{PID}_{i,2} \oplus h_1(b\text{PID}_{i,2}, \Delta T_i)$ to get the true identity RID_i for the vehicle V_i to track the vehicle. In the event of a dispute, it is necessary to reveal the identity by tracking the vehicle.
3) *Unlinkability:* Unlinkability means that the attacker cannot link the relationship between two messages sent by the same vehicle. In this article, the vehicle V_i transfers a message $\{\text{PID}_i, vpk_{\text{PID}_i}, M_i, T_i, \sigma_i\}$ to the nearby RSU. Due to the randomness of r_i in the signature σ_i, the attacker cannot associate two messages $\{\text{PID}_i, vpk_{\text{PID}_i}, M_i, T_i, \sigma_i\}$ to the same vehicle. Hence, the proposed eCLAS scheme could provide nonlinkability.
4) *Message authentication and integrity:* According to aforementioned formal security analyze, we proved that our scheme is unforgeable. And we can verify the integrity and validity of the message $\{\text{PID}_i, vpk_{\text{PID}_i}, M_i, T_i, \sigma_i\}$ by checking whether the $S_i P = W_i + h_i vpk_{\text{PID}_i}$ condition is true. Therefore, the proposed eCLAS scheme satisfies message integrity and authentication requirements.
5) *Replay attack:* In the proposed eCLAS scheme, the timestamp T_i is added the authentication message $\{\text{PID}_i, vpk_{\text{PID}_i}, M_i, T_i, \sigma_i\}$ to ensure the newness. RSU_i is able to detect replay attack through the timestamp. So, the proposed eCLAS scheme could resist replay attack.
6) *Impersonation attack:* According to the above, we proved that our scheme is unforgeable for adaptively selective message attack under the ECDLP difficulty assumption in the random oracle model. An attacker cannot fake message $\{\text{PID}_i, vpk_{\text{PID}_i}, M_i, T_i, \sigma_i\}$ to satisfy the verification equation $S_i P = W_i + h_i vpk_{\text{PID}_i}$. Therefore, our proposed eCLAS scheme is able to resist impersonation attack.
7) *Modification attack:* During the transmission of the message $\{\text{PID}_i, vpk_{\text{PID}_i}, M_i, T_i, \sigma_i\}$ over the insecure channel, any modification to the message due to the attacker or the network problem, the verifier determines whether the message is modified by detecting whether the $S_i P = W_i + h_i vpk_{\text{PID}_i}$ equation is established. Therefore, our proposed eCLAS scheme is able to resist the modification attack.
8) *Message spoofing attack:* The message sent by the vehicle is $\{\text{PID}_i, vpk_{\text{PID}_i}, M_i, T_i, \sigma_i\}$, where $\sigma_i = (U_i, S_i)$, $S_i = psk_{\text{PID}_i} + vsk_{\text{PID}_i} h_i$ and $psk_{\text{PID}_i} = aQ_{\text{ID}_i}$. Because the attacker cannot solve the ECDLP, the adversary cannot successfully forge false messages that pass the verification.
9) *Man-in-the-middle attack:* In the proposed eCLAS, the participating entities will authenticate each other. Because attackers cannot impersonate legal entities to send authenticated messages, the scheme can resist man-in-the-middle attacks.

VI. Performance Analysis

In this article, the computational evaluation method we adopted is similar to that in scheme [40]. Bilinear pairing is constructed as follows: bilinear pairing $\bar{e}$: $G_1 \times G_1 \to G_2$ are built on the security level of 80 b. G_1 is an additive group whose order is $\hat{q}$ and the generator is $\hat{p}$, which is a point on the super singular elliptic curve $\bar{E}$: $y^2 = x^3 + x \mod \hat{p}$ with an embedding degree of 2. $\hat{p}$ is a 512-b prime number and $\hat{q} = 2^{159} + 2^{17} + 1$ is a 160-b prime number. ECC, constructed at 80-b security level: G is an additive group whose order is q and the generator is a point P on a nonsingular elliptic curve $\bar{E}$: $y^2 = x^3 + ax + b \mod p$, where $a, b \in Z_p^*$, and q is a 160-b prime.

By using the C/C++ cryptographic library called MIRACL, we measure the execution time of involved cryptography operations, as shown in Table II, where the hardware platform contains Intel I7-6700 processor and 8-GB memory and runs Windows 7 operating system.

A. Security Comparison

Table III shows the comparison results, including authentication, anonymity, security against A_1, security against A_2, without pairing and signature scheme supports. Here, the entry $\surd$ indicates that the scheme meets the demand, and the entry $\times$ indicates that the scheme does not satisfy the target. It is noted that among all the schemes compared in Table III, only our

TABLE II
EXECUTION TIME OF DIFFERENT CRYPTOGRAPHIC OPERATIONS

Scheme	Authentication	Anonymity	Security against A_1	Security against A_2	Without pairing	Signature scheme supports
Cui et al.[34]	√	√	√	×	√	Aggregation,Batch Verification
Wu et al.[35]	√	×	√	√	×	Aggregation
Kamil et al.[12]	√	√	×	×	√	Aggregation
Horng et al.[6]	√	√	×	×	×	Aggregation, Batch Verification
Kumar et al.[7]	√	√	√	√	×	Aggregation
Zhong et al.[11]	√	√	√	×	×	Aggregation, Batch Verification
Thumbur et al.[36]	√	√	√	√	√	Aggregation
Xu et al.[37]	√	√	√	×	×	Aggregation
Sikarwar et al.[38]	√	√	√	√	×	Batch Verification
Xiong et al.[39]	√	√	√	√	√	Batch Verification
Our scheme	√	√	√	√	√	Aggregation, Batch Verification

TABLE III
COMPARISON OF SECURITY PROPERTIES IN SCHEMES

Notations	Definitions	Execution time (millisecond)
T_{bp}	Bilinear pairing operation	5.086
T_{sm-bp}	The scale multiplication operation based on bilinear pairing	0.694
$T_{sm-bp-s}$	The small scale scalar multiplication operation based on bilinear maps	0.0736
T_{pa-bp}	The point addition operation based on bilinear pairing	0.0018
T_{mtp}	The hash-to-point operation based on bilinear pairing	0.0992
T_{sm-ecc}	The scale multiplication operation based on elliptic curve	0.3218
$T_{sm-ecc-s}$	The small scale scalar multiplication operation related to elliptic curves	0.0246
T_{pa-ecc}	Calculating the point addition operation related to elliptic curves	0.0024
T_h	Hash operation	0.001

TABLE IV
CALCULATION OVERHEAD FOR SIGNATURE GENERATION AND AGGREGATION VERIFICATION IN DIFFERENT SCHEMES

Schemes	Message signature	Aggregation verification
Horng et al.[6]	$2T_{sm-bp} + T_{pa-bp} + T_h \approx 1.3908$	$3T_{bp} + nT_{sm-bp} + nT_{pa-bp} + nT_{mtp} + nT_h$
Kumar et al.[7]	$4T_{sm-bp} + 2T_{pa-bp} + T_{mtp} + 2T_h \approx 2.8798$	$4T_{bp} + 3nT_{sm-bp} + T_{mtp} + 3nT_h$
Zhong et al.[11]	$3T_{sm-bp} + T_{pa-bp} + T_h \approx 2.0848$	$3T_{bp} + 2nT_{sm-bp} + (2n-1)T_{pa-bp} + nT_{mtp} + nT_h$
Thumbur et al.[36]	$T_{sm-ecc} + 4T_h \approx 0.3258$	$(2n-1)T_{sm-ecc} + (3n-2)T_{pa-ecc} + (2n)T_h$
Xu et al.[37]	$T_{sm-ecc} + T_h \approx 0.3228$	$(2n-1)T_{sm-ecc} + (3n-1)T_{pa-ecc} + (2n)T_h$
Our scheme	$T_{sm-ecc} + T_h \approx 0.3228$	$(n+1)T_{sm-ecc} + (2n-1)T_{pa-ecc} + nT_h$

scheme provides better security and more functionality features as compared to other schemes.

B. Computation Cost Analysis

1) Signature Generation and Aggregate Verification cost: We compare the proposed eCLAS scheme with the related schemes [6], [7], [11], [36], [37] from the aspects of message signature generation and aggregate signature verification.

In Horng *et al.*'s scheme [6], the overheads of signature generation and aggregate signature verification are: $2T_{sm-bp} + T_{pa-bp} + T_h$ and $3T_{bp} + nT_{sm-bp} + nT_{pa-bp} + nT_{mtp} + nT_h$, respectively. In Kumar *et al.*'s scheme [7], the overheads of signature generation and aggregate signature verification are: $4T_{sm-bp} + 2T_{pa-bp} + T_{mtp} + 2T_h$ and $4T_{bp} + 3nT_{sm-bp} + T_{mtp} + 3nT_h$. The scheme of Zhong *et al.* [11] has the following overheads for signature generation and aggregate signature verification: $3T_{sm-bp} + T_{pa-bp} + T_h$ and $3T_{bp} + 2nT_{sm-bp} + (2n-1)T_{pa-bp} + nT_{mtp} + nT_h$. In the scheme of Thumbur *et al.* [36], the overheads of signature generation and aggregate signature verification are: $T_{sm-ecc} + 4T_h$ and $(2n-1)T_{sm-ecc} + (3n-2)T_{pa-ecc} + (2n)T_h$, respectively. In the scheme of Xu *et al.* [37], the overheads of signature generation and aggregate signature verification are: $T_{sm-ecc} + T_h$ and $(2n-1)T_{sm-ecc} + (3n-1)T_{pa-ecc} + (2n)T_h$, respectively.

Correspondingly, our scheme's corresponding costs are: $T_{sm-ecc} + T_h$ and $(n+1)T_{sm-ecc} + (2n-1)T_{pa-ecc} + nT_h$. We clearly list the overhead of signature generation and aggregate signature verification cost in different schemes through Table IV.

For verifying the aggregate signature of 1000 messages, Horng *et al.* [6], Kumar *et al.* [7], Zhong *et al.* [11], Thumbur *et al.* [36], Xu *et al.* [37] as well as our scheme need 811.258, 2105.4432, 1507.0562, 654.47, 654.48, and 327.9194ms, respectively. That is, our scheme has improved 59.6%, 84.4%, 78.2%, 49.89%, and 49.89% compared with aforementioned three schemes. In addition, as can be seen from Fig. 3, in the process of different numbers of aggregation signature verification, the verification cost of our solution is significantly lower than those three solutions. Consequently, our scheme can effectively aggregate the signatures of multiple messages into a relatively short signature, which makes it necessary to spend less time in the verification process.

2) Single Verification and Batch Verification Cost: To ensure the nonrepudiation of signatures adopting aggregate signature verification, we introduce the small exponent test technology into the batch verification of multiple messages. We compare it with existing solutions [6], [38], [39], [41]–[43].

In the scheme of Horng *et al.* [6], the overheads in the single signature verification and batch verification

TABLE V
CALCULATION OVERHEAD FOR SIGNATURE VERIFICATION AND BATCH VERIFICATION IN DIFFERENT SCHEMES

Schemes	Individual verification	Batch verification
Horng et al.[6]	$3T_{bp} + T_{sm-bp} + T_{pa-bp} + T_{mtp} + T_h$	$3T_{bp} + nT_{sm-bp} + 3nT_{sm-bp-s} + nT_{pa-bp} + nT_{mtp} + nT_h$
Horng et al.[41]	$2T_{bp} + 2T_{sm-bp} + T_{pa-bp} + T_{mtp} + T_h$	$2T_{bp} + 2nT_{sm-bp} + nT_{pa-bp} + nT_{mtp} + nT_h$
Liu et al.[42]	$2T_{bp} + 2T_{sm-bp} + 2T_h$	$2T_{bp} + (n+1)T_{sm-bp} + 2nT_h$
Zhong et al.[43]	$3T_{sm-ecc} + T_{pa-ecc} + 2T_h$	$(n+2)T_{sm-ecc} + 2nT_{sm-ecc-s} + (2n-1)T_{pa-ecc} + 2nT_h$
Sikarwar et al.[38]	$3T_{bp} + T_{sm-bp} + T_h$	$3T_{bp} + nT_{sm-bp} + 3nT_{pa-bp} + nT_h$
Xiong et al.[39]	$3T_{sm-ecc} + 2T_{pa-ecc} + 2T_h$	$(n+2)T_{sm-ecc} + nT_{sm-ecc-s} + (2n-1)T_{pa-ecc} + 2nT_h$
Our scheme	$2T_{sm-ecc} + T_{pa-ecc} + T_h$	$(n+1)T_{sm-ecc} + nT_{sm-ecc-s} + (2n-1)T_{pa-ecc} + nT_h$

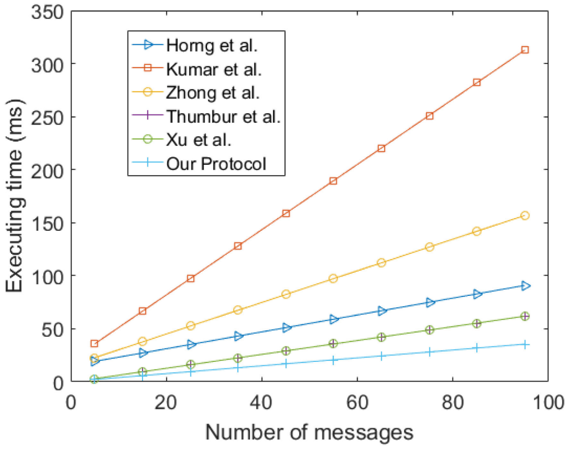

Fig. 3. Calculation delay of signature aggregation verification process in different schemes.

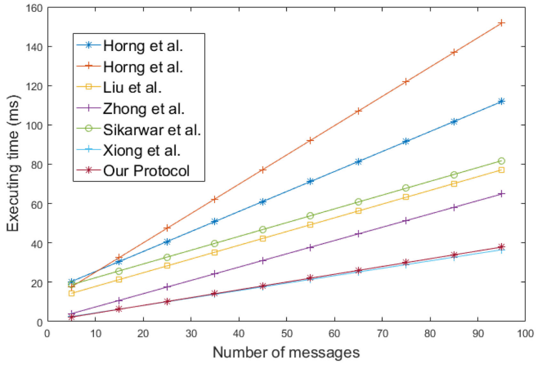

Fig. 4. Calculation delay of batch verification process in different schemes.

processes are: $3T_{bp} + T_{sm-bp} + T_{pa-bp} + T_{mtp} + T_h$ and $3T_{bp} + nT_{sm-bp} + 3nT_{sm-bp-s} + nT_{pa-bp} + nT_{mtp} + nT_h$, respectively. In the scheme of Horng *et al.* [41], the overheads in the single signature verification and batch verification processes are: $2T_{bp} + 2T_{sm-bp} + T_{pa-bp} + T_{mtp} + T_h$ and $2T_{bp} + 2nT_{sm-bp} + nT_{pa-bp} + nT_{mtp} + nT_h$. In the scheme of Liu *et al.* [42], the overheads in the single signature verification and batch verification processes are: $2T_{bp} + 2T_{sm-bp} + 2T_h$ and $2T_{bp} + (n+1)T_{sm-bp} + 2nT_h$. In the scheme of Zhong *et al.* [43], the overheads in the single signature verification and batch verification processes are: $3T_{sm-ecc} + T_{pa-ecc} + 2T_h$ and $(n+2)T_{sm-ecc} + 2nT_{sm-ecc-s} + (2n-1)T_{pa-ecc} + 2nT_h$, respectively.

TABLE VI
COMPARISON OF COMMUNICATION COST

Scheme	Communication Costs
Horng et al.[6]	$4\lvert G_1\rvert + \lvert Z_p^*\rvert + 4 = 536$ bytes
Kumar et al.[7]	$4\lvert G_1\rvert + \lvert Z_p^*\rvert = 532$ bytes
Zhong et al.[11]	$3\lvert G_1\rvert + \lvert Z_p^*\rvert + 4 = 408$ bytes
Thumbur et al.[36]	$4\lvert G\rvert + \lvert Z_p^*\rvert + 4 = 184$ bytes
Xu et al. [37]	$3\lvert G\rvert + \lvert Z_p^*\rvert + 4 = 144$ bytes
Sikarwar et al.[38]	$3\lvert G_1\rvert + 4 = 388$ bytes
Xiong et al.[39]	$3\lvert G\rvert + \lvert Z_p^*\rvert + 8 = 148$ bytes
Our scheme	$3\lvert G\rvert + \lvert Z_p^*\rvert + 4 = 144$ bytes

In the scheme of Sikarwar *et al.* [38], the overheads in the single signature verification and batch verification processes are: $3T_{bp} + T_{sm-bp} + T_h$ and $3T_{bp} + nT_{sm-bp} + (3n)T_{pa-bp} + nT_h$, respectively. In the scheme of Xiong *et al.* [39], the overheads in the single signature verification and batch verification processes are: $3T_{sm-ecc} + 2T_{pa-ecc} + 2T_h$ and $(n+2)T_{sm-ecc} + nT_{sm-ecc-s} + (2n-1)T_{pa-ecc} + 2nT_h$, respectively.

Correspondingly, our scheme's costs are: $2T_{sm-ecc} + T_{pa-ecc} + T_h$ and $(n+1)T_{sm-ecc} + nT_{sm-ecc-s} + (2n-1)T_{pa-ecc} + nT_h$. In Table V, we lists the verification delays of single verification and batch verification cost in different schemes.

For batch verifying the signatures of 100 messages, Horng *et al.* [6], Horng *et al.* [41], Liu *et al.* [42], Zhong *et al.* [43], Sikarwar *et al.* [38], Xiong *et al.* [39], and our schemes need 116.638, 159.172, 80.466, 38.4212, 85.298, 68.221, and 35.5394ms, respectively. That is, our scheme has improved 69.6%, 77.7%, 55.8%, 7.46%, 58.3%, and 47.83% compared with aforementioned schemes. Fig. 4 shows the computation cost of the batch verification process in different schemes. As we can see from Fig. 4, our scheme has a significant advantage over the scheme [6], [39], [41]– [43], which can verify more message signatures in the same time, effectively reduce the time delay of the verification process, and improve the efficiency of the batch process.

C. Communication Cost Analysis

As $\overline{p}$ and p are 64 and 20 B, the sizes of the elements in G_1 and G are $64 \times 2 = 128$ and $20 \times 2 = 40$ B, respectively. Set the size of timestamp be 4 B and the output of the general hash function be 20 B. Here, we only consider the size of transmitted from vehicles. The specific computation costs are shown in Table VI.

TABLE VII
STORAGE OVERHEAD

Entity	Stored content	Notations	Storage overhead (bytes)
KGC	master private key	a	20
TRA	master private key, Vehicles' real identities	b, RID_i	$40 * k + 20$
Vehicle	private key, Vehicles' pseudonym identities	$\{vsk_{PID_i}, psk_{PID_i}\}$, PID_i	$84 * Q + 40$

A vehicle V_i in Horng *et al.* [6] transmits an anonymous identity $\text{ID}_i = \{\text{ID}_{i,1}, \text{ID}_{i,2}\}$ for $\text{ID}_{i,1} \in G_1$ and $\text{ID}_{i,2} \in Z_p^*$, a public key $vpk_i \in G_1$, a time-stamp t_i, and a signature $\sigma_i = (R_i, S_i) \in G_1$ to the RSU. Therefore, the total communication cost incurred from Horng *et al.*' scheme [6] is approximately equal to $4|G_1| + |Z_p^*| + 4 = 4 \times 128 + +40 + 4 = 536$ B.

In the scheme of Kumar *et al.*'s [7], a vehicle sends an anonymous identity $PS_j = \{PS_{1,j}, PS_{2,j}\}$ for $PS_{1,j} \in G_1$ and $PS_{2,j} \in Z_p^*$, a public key $P_i \in G_1$, and a signature $\sigma_i = (U_i, V_{ijk}) \in G_1$ to the verifier; in summary, Kumar *et al.*' scheme [7] incurs a total communication cost, which is approximately equal to $4|G_1| + |Z_p^*| = 4 \times 128 + +40 = 532$ B. In Zhong *et al.*' scheme [11], a vehicle sends its pseudonym identity $\text{PID}_i = \{\text{PID}_{i,1}, \text{PID}_{i,2}\}$ for $\text{PID}_{i,1} \in G_1$ and $\text{PID}_{i,2} \in Z_p^*$, a public key $vpk_i \in G_1$, a time-stamp t_i, and a signature $\sigma_i = (R_i, T_i) \in G_1$ to the RSU. Thus, Zhong *et al.*' scheme [11] incurs a total communication cost, which is about $3|G_1| + |Z_p^*| + 4 = 3 \times 128 + 20 + 4 = 408$ B. In Thumbur *et al.*' scheme [36], a vehicle sends $\{P_{\text{ID}}{}^{i}, M_i, vpk_{\text{PID}i}, \sigma_i, T_i\}$. Thus, Thumbur *et al.*' scheme incurs a total communication cost, which is $4|G| + |Z_p^*| + 4 = 184$ B. In Xu *et al.*' scheme [37], a vehicle sends $\{\text{PID}_i, M_i, t_i, \sigma_i\}$. Thus, Xu *et al.*' scheme incurs a total communication cost, which is $3|G| + |Z_p^*| + 4 = 144$ B. In Sikarwar *et al.*' scheme [38], a vehicle transmits the message tuple $\{P_{\text{ID}}{}^{i}, M_i, \text{sign}_i, T_i\}$. Therefore, the communication cost is approximately equal to $3|G_1| + 4 = 388$ B. In Xiong *et al.*' scheme [39], a vehicle transmits the message tuple $\{A_{j,i}, M_{j,i}, \text{PID}_{j,i}, T_{j,i}, S_{pubj}, \beta_{j,i}, t_{j,i}\}$. Because $\{A_{j,i}, S_{pubj}, \beta_{j,i}\} \in G$, $t_{j,i}$ is the timestamp and $\text{PID}_{j,i} \in Z_p^*$, therefore, the communication cost of Xiong *et al.*' scheme is $40 \times 3 + 20 + 8 = 148$ B.

In the proposed eCLAS scheme, the vehicle transmits the anonymous identity and signature $\{\text{PID}_i, vpk_{\text{PID}_i}, M_i, T_i, \sigma_i\}$, where $\sigma_i = (U_i, S_i)$, PID_i, vpk_{PID_i}, $U_i \in G$, T_i is the timestamp. S_i equals a hash operation result. As a result, the communication cost of our scheme is approximately equal to $40 \times 3 + 20 + 4 = 144$ B. Consequently, the proposed eCLAS scheme incurs a much lower communication cost and better in bandwidth limited VANETs than Horng *et al.* [6], Kumar *et al.* [7], Zhong *et al.* [11], and Sikarwar *et al.* [38].

D. Storage Cost Analysis

The following is our analysis of the storage cost required by different participating entities in the proposed eCLAS scheme. The corresponding contents are listed in Table VII.

In our scheme, both the AS and the RSU are only responsible for authentication, and because there is no need to store any private information at the AS and the RSU, the storage overhead at the AS and the RSU is 0 B.

What KGC needs to store is its own master key a. Because $a \in Z_p^*$, the storage overhead of KGC is only 20 B. TRA needs to store not only its own master key b, but also the real identities RID_i of the vehicles to check the registered vehicle, that is, to ensure that the vehicle has not been registered or is not on the blacklist. Because $b \in Z_p^*$ and $\text{RID}_i \in G$. Assuming that the number of registered vehicles is k, the storage overhead of TRA is $40 * k + 20$ B.

The vehicle needs to store its own private key $\{vsk_{\text{PID}_i}, psk_{\text{PID}_i}\}$ and pseudonyms PID_i. It can be seen the private key consists of two parts, that is, the private key $vsk_{\text{PID}_i} \in Z_p^*$ selected by the vehicle itself and the private key $psk_{\text{PID}_i} \in Z_p^*$ assigned by KGC. Besides, the vehicle needs to store many pseudonyms PID_i in advance, the pseudonyms are in the form of $\text{PID}_i = \{\text{PID}_{i,1}, \text{PID}_{i,2}, \Delta T_i\}$, where $\text{PID}_{i,1} \in G$ and $\text{PID}_{i,2} \in G$. Assuming that the stored number is Q. Because the storage overhead at the vehicle end is $84 * Q + 40$ B.

VII. CONCLUSION

In this article, an improved certificateless aggregation signature scheme for VANETs based on ECC and without bilinear pairing operations has been proposed, which reduced the length of the message signature and the timing overhead of the verification process. The proposed eCLAS scheme ensured the security of type I and type II attackers under the hardness assumption of ECDLP in the random oracle model. Moreover, the detailed analysis showed that the proposed eCLAS scheme can meet the security requirements in VANETs. Additionally, we compared and analyzed existing schemes from the perspective of aggregate signature and batch verification. The results verified that our scheme can effectively reduce delays and improve authentication efficiency. In the future work, we will go on designing a novel scheme for VANETs authentication in 5G environment.

REFERENCES

[1] R. G. Engoulou, M. Bellaïche, S. Pierre, and A. Quintero, "VANET security surveys," *Comput. Commun.*, vol. 44, pp. 1–13, 2014.

[2] J. Kang, R. Yu, X. Huang, and Y. Zhang, "Privacy-preserved pseudonym scheme for fog computing supported Internet of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2627–2637, Aug. 2018.

[3] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (VANETs): Status, results, and challenges," *Telecommun. Syst.*, vol. 50, no. 4, pp. 217–241, 2012.

[4] W. Hathal, H. Cruickshank, Z. Sun, and C. Maple, "Certificateless and lightweight authentication scheme for vehicular communication networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 16110–16125, Dec. 2020.

[5] J. Zhang, H. Zhong, J. Cui, Y. Xu, and L. Liu, "SMAKA: Secure many-to-many authentication and key agreement scheme for vehicular networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 1810–1824, 2020.

[6] S.-J. Horng, S.-F. Tzeng, P.-H. Huang, X. Wang, T. Li, and M. K. Khan, "An efficient certificateless aggregate signature with conditional privacy-preserving for vehicular sensor networks," *Inf. Sci.*, vol. 317, pp. 48–66, 2015.

[7] P. Kumar, S. Kumari, V. Sharma, X. Li, A. K. Sangaiah, and S. H. Islam, “Secure CLS and CL-AS schemes designed for VANETs,” *J. Supercomput.*, vol. 75, no. 6, pp. 3076–3098, 2019.

[8] J. Zhang, J. Cui, H. Zhong, I. Bolodurina, and L. Liu, “Intelligent drone-assisted anonymous authentication and key agreement for 5G/B5G vehicular ad-hoc networks,” *IEEE Trans. Netw. Sci. Eng.*, to be published, doi: 10.1109/TNSE.2020.3029784.

[9] Y. Li, Q. Luo, J. Liu, H. Guo, and N. Kato, “TSP security in intelligent and connected vehicles: Challenges and solutions,” *IEEE Wireless Commun.*, vol. 26, no. 3, pp. 125–131, Jun. 2019.

[10] M. Obaidat, M. Khodjaeva, J. Holst, and M. B. Zid, “Security and privacy challenges in vehicular ad hoc networks,” in *Connected Vehicles in the Internet of Things*. Berlin, Germany: Springer, 2020, pp. 223–251.

[11] H. Zhong, S. Han, J. Cui, J. Zhang, and Y. Xu, “Privacy-preserving authentication scheme with full aggregation in VANET,” *Inf. Sci.*, vol. 476, pp. 211–221, 2019.

[12] I. A. Kamil and S. O. Ogundoyin, “An improved certificateless aggregate signature scheme without bilinear pairings for vehicular ad hoc networks,” *J. Inf. Secur. Appl.*, vol. 44, pp. 184–200, 2019.

[13] K.-A. Shim, “CPAS: An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks,” *IEEE Trans. Veh. Technol.*, vol. 61, no. 4, pp. 1874–1883, May 2012.

[14] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, “Aggregate and verifiably encrypted signatures from bilinear maps,” in *Proc. Int. Conf. Theory Appl. Cryptogr. Techn.*, 2003, pp. 416–432.

[15] C. Gentry and Z. Ramzan, “Identity-based aggregate signatures,” in *Proc. Int. Workshop Public Key Cryptogr.*, 2006, pp. 257–273.

[16] W. Yi-ling, M. Jian-feng, and W. Chao, “New ID-based aggregate signature scheme,” *Comput. Sci.*, vol. 38, pp. 54–57, 2011.

[17] X. Cheng, J. Liu, and X. Wang, “An ID-based aggregate signature scheme from M-torsion groups,” *J. Xidian Univ.*, vol. 32, no. 3, pp. 427–431, 2005.

[18] Y. Yu, X. Zheng, and H. Sun, “A new ID-based aggregate signature with constant pairing operations,” in *Proc. 2nd Int. Conf. Netw. Secur. Wireless Commun., Trusted Comput.*, 2010, vol. 2, pp. 188–191.

[19] 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 J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 1191–1204, Jun. 2020.

[20] J. Song, H. Kim, S. Lee, and H. Yoon, “Security enhancement in ad hoc network with ID-based cryptosystem,” in *Proc. 7th Int. Conf. Adv. Commun. Technol.*, 2005, vol. 1, pp. 372–376.

[21] 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 Trans. Dependable Secure Comput.*, vol. 18, no. 2, pp. 722–735, Mar./Apr. 2021.

[22] S. S. Al-Riyami and K. G. Paterson, “Certificateless public key cryptography,” in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, 2003, pp. 452–473.

[23] D. H. Yum and P. J. Lee, “Generic construction of certificateless signature,” in *Proc. Australas. Conf. Inf. Secur. Privacy*, 2004, pp. 200–211.

[24] Z. Zhang, D. S. Wong, J. Xu, and D. Feng, “Certificateless public-key signature: Security model and efficient construction,” in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.*, 2006, pp. 293–308.

[25] Z. Gong, Y. Long, X. Hong, and K. Chen, “Two certificateless aggregate signatures from bilinear maps,” in *Proc. 8th ACIS Int. Conf. Softw. Eng., Artif. Intell., Netw., Parallel/Distrib. Comput.*, 2007, vol. 3, pp. 188–193.

[26] H. Xiong, Z. Guan, Z. Chen, and F. Li, “An efficient certificateless aggregate signature with constant pairing computations,” *Inf. Sci.*, vol. 219, pp. 225–235, 2013.

[27] I. Ali, M. Gervais, E. Ahene, and F. Li, “A blockchain-based certificateless public key signature scheme for vehicle-to-infrastructure communication in VANETs,” *J. Syst. Architecture*, vol. 99, 2019, Art. no. 101636.

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

[29] I. A. Kamil and S. O. Ogundoyin, “On the security of privacy-preserving authentication scheme with full aggregation in vehicular ad hoc network,” *Secur. Privacy*, vol. 3, no. 3, 2020, Art. no. e104.

[30] Z. Xu, D. He, N. Kumar, and K.-K. R. Choo, “Efficient certificateless aggregate signature scheme for performing secure routing in VANETs,” *Secur. Commun. Netw.*, vol. 2020, 2020, Art. no. 5276813.

[31] L. Armstrong, “Dedicated short range communications (DSRC) home,” Standards, 2002.

[32] J. Huang, L. Yeh, and H. Chien, “ABAKA: An anonymous batch authenticated and key agreement scheme for value-added services in vehicular ad hoc networks,” *IEEE Trans. Veh. Technol.*, vol. 60, no. 1, pp. 248–262, Jan. 2011.

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

[34] J. Cui, J. Zhang, H. Zhong, R. Shi, and Y. Xu, “An efficient certificateless aggregate signature without pairings for vehicular ad hoc networks,” *Inf. Sci.*, vol. 451, pp. 1–15, 2018.

[35] L. Wu, Z. Xu, D. He, and X. Wang, “New certificateless aggregate signature scheme for healthcare multimedia social network on cloud environment,” *Secur. Commun. Netw.*, vol. 2018, 2018, Art. no. 2595273.

[36] G. Thumbur, G. S. Rao, P. V. Reddy, N. Gayathri, D. K. Reddy, and M. Padmavathamma, “Efficient and secure certificateless aggregate signature-based authentication scheme for vehicular ad hoc networks,” *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1908–1920, Feb. 2020.

[37] G. Xu *et al.*, “A security-enhanced certificateless aggregate signature authentication protocol for InVANETs,” *IEEE Netw.*, vol. 34, no. 2, pp. 22–29, Mar./Apr. 2020.

[38] H. Sikarwar, A. Nahar, and D. Das, “LABVS: Lightweight authentication and batch verification scheme for universal internet of vehicles (UIoV),” in *Proc. IEEE 91st Veh. Technol. Conf.*, 2020, pp. 1–6.

[39] W. Xiong, R. Wang, Y. Wang, F. Zhou, and X. Luo, “CPPA-D: Efficient conditional privacy-preserving authentication scheme with double-insurance in VANETs,” *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3456–3468, Apr. 2021.

[40] H. Zhong, B. Huang, J. Cui, Y. Xu, and L. Liu, “Conditional privacy-preserving authentication using registration list in vehicular ad hoc networks,” *IEEE Access*, vol. 6, pp. 2241–2250, 2017.

[41] S.-J. Horng *et al.*, “b-SPECS+: Batch verification for secure pseudonymous authentication in VANET,” *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 11, pp. 1860–1875, Nov. 2013.

[42] Y. Liu, Z. He, S. Zhao, and L. Wang, “An efficient anonymous authentication protocol using batch operations for VANETs,” *Multimedia Tools Appl.*, vol. 75, no. 24, pp. 17689–17709, 2016.

[43] H. Zhong, J. Wen, J. Cui, and S. Zhang, “Efficient conditional privacy-preserving and authentication scheme for secure service provision in VANET,” *Tsinghua Sci. Technol.*, vol. 21, no. 6, pp. 620–629, 2016.