Efficient and Secure Certificateless Aggregate Signature-Based Authentication Scheme for Vehicular *Ad Hoc* Networks

Gowri Thumbur, *Senior Member, IEEE*, G. Srinivasa Rao, P. Vasudeva Reddy, N. B. Gayathri, D. V. R. Koti Reddy, *Member, IEEE*, and M. Padmavathamma

***Abstract*—In recent years, IoT has opened new opportunities for the development of various industries to improve people's lives. Vehicular *ad hoc* network (VANET) uses IoT applications for secure communication among the vehicles and to improve road safety and traffic management. In VANETS, the authentication of the vehicular access control is a crucial security service for both intervehicle and vehicle–roadside unit communications. Another criteria is all the messages should be unaltered in the delivery. Meanwhile, vehicles have to be prevented from the misuse of private information and the attacks on their privacy. Also, limited bandwidth, high mobility and density of vehicles, and scalability are few other challenges in VANETS. A number of research works are focusing on providing the anonymous authentication with preserved privacy and security in VANETS. In this article, we proposed a new certificateless aggregate signature-based authentication scheme for VANETS. Our scheme avoids the complex certificate management problem from public-key infrastructure and key escrow problem from an identity-based framework. Also, aggregate signature aggregates various individual signatures on different messages from different vehicles into a single signature, which in turn results in the reduction of verification time and storage space at the roadside unit. Our scheme can prevent malicious vehicles from disrupting the security features of VANETS. Moreover, our scheme does not use the pairing operation, which is the most expensive operation than others in modern cryptography, thus significantly reduces the computation overhead. Security and performance analysis shows that our scheme is more secure and efficient than current schemes.**

***Index Terms*—Aggregate signature, authentication, elliptic curve cryptography, intelligent transportation system, vehicular *ad hoc* networks (VANETs).**

I. Introduction

IN THE era of industrial 4.0, integrating existing and new technology with Internet-of-Things systems offers a beneficial impact. IoT is promising in transforming many fields, such as medicine, urban planning, power, smart transportations, and so on. Vehicular *ad hoc* network (VANET) uses IoT applications and intelligent transportation mechanism to create a space for secure communication among the vehicles [1]. A general reference model for VANETs is given in Fig. 1. It consists of vehicles installed with onboard units (OBUs), roadside units (RSUs) and a trusted authority (TA), and applications servers (ASs). RSUs are set up along the roadside and receive the information from vehicles [2], [3]. VANETs provide three types of communications, such as vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and infrastructure-to-infrastructure (I2I) communications. V2V communications are done through open dedicated short-range radio signals (DSRSs) while V2I and I2I communications are done through secure channels [4], [5]. Through VANETs, each vehicle communicates using OBUs and broadcast traffic-related information, such as positions, speed, current time, traffic, road conditions, etc., to a nearby vehicle and RSU [4]–[6].

Though VANETS has many advantages, it focuses on some security challenges and privacy issues during the sharing of information among the vehicles. Because of the open wireless network of VANETs, any attacker could send false information to the RSUs or other vehicles, which may lead to potential traffic problems. To achieve road safety, it is necessary to verify the authentication and integrity of messages before they can be deemed reliable [2]–[7]. Advanced cryptographic techniques can be applied to messages to provide security. Thus, we must resolve some security and privacy issues for V2I communications in VANETS: message integrity, source authentication, traceability, and unlinkability. A considerable amount of work has been done in constructing the authentication schemes for VANETs based on digital signatures in different cryptographic frameworks, such as PKI-based, ID-based, and Certificateless-based setting. We should also consider the efficiency in the design of a feasible authentication scheme for efficient communication in VANETs because some of the entities in VANETs, such as RSUs and OBUs, have limited computational storage capacity. The concept of aggregate signature was proposed by Boneh *et al.* [8] in 2003, which allows individual signature on different messages from different vehicles to be aggregated into one short signature. Such signatures improve the computational as well as communication cost and storage efficiency. Due to the

Manuscript received April 30, 2020; revised June 26, 2020 and July 13, 2020; accepted August 7, 2020. Date of publication August 25, 2020; date of current version January 22, 2021. *(Corresponding author: P. Vasudeva Reddy.)*

Gowri Thumbur is with the Department of Electronics and Communication Engineering, GIT, GITAM University, Visakhapatnam 530045, India.

G. Srinivasa Rao, P. Vasudeva Reddy, and N. B. Gayathri are with the Department of Engineering Mathematics, Andhra University, Visakhapatnam 530003, India (e-mail: vasucrypto@andhrauniversity.edu.in).

D. V. R. Koti Reddy is with the Department of Instrument Technology, Andhra University, Visakhapatnam 530003, India.

M. Padmavathamma is with the Department of Computer Science, S. V. University, Tirupati 517502, India.

Digital Object Identifier 10.1109/JIOT.2020.3019304

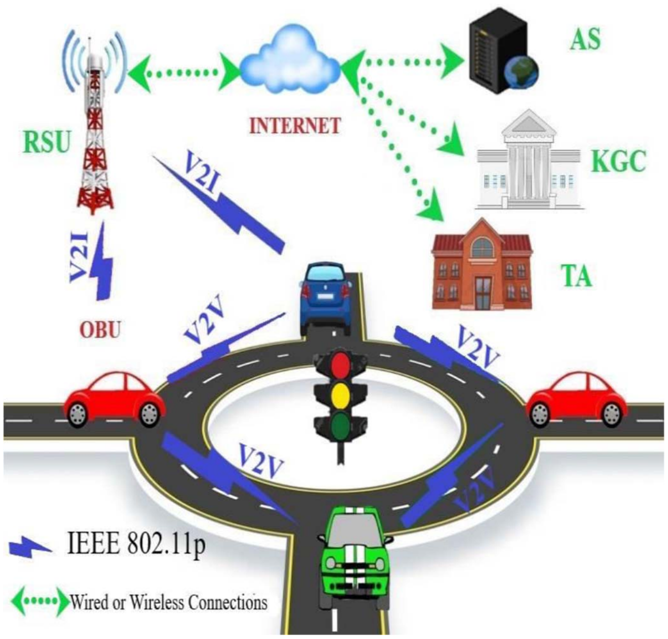

Fig. 1. VANET architecture.

advantages of aggregation, many aggregate signature schemes have been proposed [9]–[13].

Recently, numerous works have been done on the security and privacy requirements in VANETS along with the aggregation concept. But, one cannot adopt any of the above schemes directly for VANETS because these schemes are not fully secure and suffer from poor performance in terms of computations, communication overhead, and high storage. In this regard, it is necessary to design an efficient and secure authentication scheme for VANETS. We, therefore, propose an authentication scheme for VANETS in a CL-based setting.

A. Motivation

Many certificateless aggregate signature (CLAS)-based authentication schemes for VANETs are constructed with expensive pairing operations. The cost for computation of one pairing operation is more than the cost of computation of a scalar multiplication in elliptic curves [14], [15]. Compared to RSA, ECC can achieve higher security levels with smaller keys, i.e., ECC with 224 bit keys achieves the same security level as RSA with 2048 bit keys. Thus, ECC improves the computational and bandwidth efficiency of the system. Also, in VANETs, RSUs need to verify a large number of messages received from the surrounding vehicles, which results in high verification cost and time. To verify a large number of messages, to reduce the computational cost and time in the verification process, the aggregation technique is the best practice. Hence, we adopt the aggregation technique for VANET-based applications to reduce computation and communication cost. This motivated us to design a pairing-free CLAS-based authentication scheme for VANETS.

B. Our Contributions

The main contributions are summarized as follows.

1) We constructed a CLAS-based authentication scheme for secure communication in VANETS.
2) The proposed CLAS scheme for VANETS does not use the expensive pairing operations so that it improves the computational efficiency of the system.
3) RSU aggregates all the signatures/messages received from the nearby vehicles into a single signature so that AS/RSU can confirm that only the registered vehicles sign the corresponding messages. Hence, the verification cost and total signature size are reduced.
4) The proposed CLAS scheme is secure and unforgeable against the potential adversaries $\mathcal{A}dv_1$ and $\mathcal{A}dv_2$ under the elliptic curve discrete logarithm problem (ECDLP) assumption.
5) Our CLAS scheme meets all the standard security requirements for VANETS.
6) Compared with other CLAS-based authentication schemes for VANETs, our scheme is efficient in computational and communicational points of view.

C. Organization of This Article

The remainder of this article is arranged as follows. Section II presents the related work. Section III presents some preliminaries, including ECC and ECDLP, VANET system model, and framework and security model of the proposed scheme. Section IV presents our CLAS-based authentication scheme for V2V and V2I communications in VANETS and its security analysis. Section V presents efficiency analysis. Finally, Section VI presents the conclusions of this article.

II. RELATED WORK

In VANETs, for V2I communication [4]–[6], [16], each RSU has to verify a large number of signed messages in the scenario of high-density traffic. Many authentication schemes have appeared in the PKI-based framework [17], [18]. In the PKI-based setting, a certificate authority (CA) is needed to manage the identities/public keys of vehicles. The use of certificates, provided by CA, increases the storage, computation, and communication burden in PKI-based schemes [5].

To solve the problems in PKI-based schemes for VANETs, many researchers put their efforts to design authentication schemes for VANETs in ID-based setting [19]–[27]. In these schemes, to provide privacy, the identifiable identity is concealed with the assist of pseudonym. But due to the inherent key escrow problem and pseudonym management overhead, the ID-based schemes are limited only to private networks [22]. To overcome the aforementioned difficulties in PKI-based setting and ID-based setting, many certificateless-based authentication schemes for VANETs have been proposed in [28]–[44]. In 2015, Horng *et al.* [28] introduced a CLAS scheme for vehicle sensor networks to provide conditional privacy. But, this scheme is insecure due to malicious key generation center (KGC) attacks [29]. In 2015, Malhi and Batra [30] proposed pairing-based CLS and CLAS schemes for VANETS under the CDH assumption. However, this scheme is insecure due to Kumar and Sharma [31] and presented an improvement in [31]. In 2018, Yang *et al.* [32] presented a cryptanalysis on Kumar and Sharma scheme [31]. In 2017, Liu *et al.* [33] applied CLAS-based authentication scheme for the IoT environment. In 2018, Kumar *et al.* [34] proposed a pairing-based CLAS scheme for the healthcare wireless sensor network system. In 2019, Zhan and Wang [35] cryptanalized the Kumar *et al.* scheme [34]. In 2018, Gayathri *et al.* [36] proposed an efficient pairing-free CLS scheme with batch verification for VANETs. In 2018, Cui *et al.* [4] proposed an ECC-based CLAS for secure V2I communication in VANETs. In the same year, Wang *et al.* [37] proposed a CLAS scheme for Vanets. However, in 2019, Hu *et al.* [38] presented security analysis on Wang *et al.* scheme [37]. In 2019, Kumar *et al.* [6] designed a CLS and CLAS schemes for VANETS. This scheme uses pairings over ECs and the security is based on CDHP. In 2019, Zhong *et al.* [39] constructed an efficient CLAS-based authentication for V2I communication in VANETs. However, Kamil and Ogundoyin [40] presented two concrete attacks on Zhong *et al.* scheme [39] and proposed an improved CLAS scheme. In 2019, Kamil and Ogundoyin [41] proved that Cui *et al.* [4] CLS and CLAS schemes are not secure against type II adversary in the random oracle model and also they proposed an improved pairing free CLAS scheme for VANETs with the assumption that the ECDLP is hard. Very recently, in 2019, Zhao *et al.* [42] showed that Kamil and Ogundoyin [41] CLS and CLAS-based authentication schemes are not secure against the forgery attack and also presented new CLS and CLAS schemes based on the hardness of ECDLP. However, in their scheme [42], the construction of CLS is not correct. For example, vehicle V_i calculates $\Phi_\lambda = r_\lambda + h_\lambda(\delta_\lambda x_\lambda + s_\lambda)$. But the random value r_λ is chosen by KGC and is not known to the vehicle V_i. Hence, vehicle V_i cannot calculate Φ_λ. Recently, in 2020, Mei *et al.* [43] proposed a CLAS with conditional privacy preservation in the Internet of vehicles using bilinear pairings. This scheme achieves full aggregation. In the same year, Xu *et al.* [44] presented an efficient CLAS for performing secure routing in VANETs. The security of Xu *et al.* [44] scheme is based on CDH assumption in the random oracle model.

III. PRELIMINARIES

A. Elliptic Curve Group and ECDLP

Let $E(F_p)$ be the set of all elliptic curve points over a finite field $F_p, p > 3$, defined by $y^2 = (x^3 + ax + b) \bmod p$, $a, b \in F_P$ and $4a^3 + 27b^2 \neq 0$. The additive elliptic curve group $G = \{(x, y) \in E(F_p) : x, y \in F_p\} \cup \{o\}$, where O is point at infinity. G forms a cyclic group under addition operation $R = P + Q$, for $P, Q \in G$ by the chord-and-tangent rule [14], [15], [22]. The scalar multiplication is defined as $kP = P + P + \cdots P(k \text{ times})$.

Elliptic Curve Discrete Logarithm Problem: For a given $P, Q \in G$, the ECDLP is to find $x \in Z_q^*, \ni Q = xP$.

B. System Architecture

The proposed VANET model consists of five entities: 1) TA; 2) a KGC; 3) authentication AS; 4) RSUs; and 5) vehicles with OBUs. Our scheme has upper and lower level communications. The communications between TA, KGC, and RSU are appeared in the upper level; whereas the communication between V2V and V to RSU appeared in the lower level. Upper level communication is performed through secure wired networks such as transport layer security protocol and lower level communication is performed through a DSRC (IEEE 802.11P) protocol. The detailed description of each entity is as follows.

1) *Trust Authority:* It is completely a TA in VANETs. TA is responsible for system initialization, registration of RSUs, and vehicles. TAs connect with RSUs through a secure channel. TA alone knows the real identities (*RID*) and in case of necessary, the TA will trace the *RID* from the corresponding pseudoidentities (PIDs) and no other party can trace this *RID*.
2) *Key Generation Centre:* It is a trusted third party independent of TA and is responsible to generate partial private keys for vehicles.
3) *Application Server:* It collects traffic-related information from RSUs and communicates with TA and RSU using a secure wired connection.
4) *Roadside Units:* It is a wireless communication device installed along the roadside to manage the communication among OBUs within its communication range using the DSRC protocol.
5) *Vehicle:* Vehicle is installed with OBU that broadcasts traffic-related information, such as traffic condition, location, vehicle direction, current line, etc., using the DSRC protocol.

The following assumptions are made in designing our scheme. The entities TA, KGC, and AS are fully independent and trusted and they will not be compromised and do not collude with attackers. RSU is a honest-but-curious entity and vehicles are untrustworthy. TA, KGC, and AS have sufficient computing and storage capabilities. The OBUs have very limited processing, computing, storage, and battery power while RSUs have more computing and battery power than OBUs. Each vehicle has a tamper-proof-protected hardware device that prevents the intruder to extract data stored in it. Each vehicle has a GPS facility. Fig. 2 explains the steps involved in the proposed authentication scheme for VANETS.

1) Vehicle registration with TA.
2) TA generates pseudoidentity and preloads in vehicles OBU.
3) Vehicle requests for a partial private key.
4) KGC generates partial private key.
5) Vehicle generates public/secret key pair.

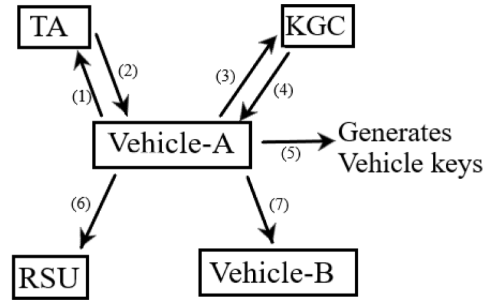

Fig. 2. Steps involved in the proposed authentication system for VANETS.

TABLE I
NOTATIONS AND THEIR MEANINGS

Notation	Meaning
TA	Trust Authority
KGC	Key Generation Centre
RSU	Road Side Unit
OBU	On Board Unit
G	Cyclic group of prime order q
$params$	System Parameters
V_i	i^{th} Vehicle
(T_{Pub}, b)	Public and private key of TA
(P_{Pub}, s)	Public and private key pair of KGC
RID_i	Real identity of the vehicle V_i
PID_i	Pseudonym identity of the vehicle V_i
psk_{PID_i}	Partial private key of the vehicle V_i
vpk_{PID_i}	Public key of the i^{th} vehicle
vsk_{PID_i}	Private key of the i^{th} vehicle
T_i	Current time stamp
ΔT_i	Valid time period of pseudo identity
H_1, H_2, H_3	Cryptographic one way hash functions
Adv_1, Adv_2	Type-I and Type-II adversaries
ξ	An algorithm which solves ECDLP
σ	Signature on a message.
ECDLP	Elliptic Curve Discrete Logarithm Problem
ID-based	Identity-based
IDBV	Identity-based Batch Verification
ECC	Elliptic Curve Cryptography

6) Vehicle-to-infrastructure (V2I) communication.
7) Vehicle-to-vehicle (V2V) communication.

The notations that are used throughout this article are tabulated in Table I.

C. Scheme Framework

Our CLAS authentication scheme consists of the following seven algorithms.

1) *System Initialization (Set Up):* TA and KGC run this algorithm with a security parameter $\lambda \in Z^+$ as input. It outputs (P_{Pub}, s) and publishes *params*.
2) *Pseudo Id Gen:* TA run this algorithm by taking $V_i^{\prime}sRID_i$ as input and outputs its PID_i.
3) *Partial Private Key Gen:* KGC runs this algorithm that takes $V_i^{\prime}sPID_i$ as input and produces respective psk_{PID_i}.
4) *Set Secret Value:* V_i takes a random value and sets this value as secret key of the vehicle.
5) *Vehicle Key Generation:* V_i executes this algorithm and generates $(vpk_{PID_i}, vsk_{PID_i})$.
6) *Sign Gen:* V_i executes this algorithm, takes $m_i \in \{0,1\}^*, psk_{PID_i}, (vpk_{PID_i}, vsk_{PID_i})$, and its PID_i as input and outputs σ_i.
7) *Sign Verification:* This algorithm is run by either V_i or RSU by taking $params, PID_i, m_i \in \{0,1\}^*$, with current timestamp, σ_i as input and outputs true if the signature is valid and false otherwise.
8) *Aggregation:* Aggregation is performed by RSU by taking various $(\sigma_i)_{i=1 \text{ to } n}$ from different $(V_i)_{i=1 \text{ to } n}$ with $(PID_i)_{i=1 \text{ to } n}$ and their respective $(vpk_{PID_i})_{i=1 \text{ to } n}$ and generate the aggregate signature σ_{agg} for $(m_i)_{i=1 \text{ to } n}$.
9) *Aggregation Verification:* Aggregate verification is performed by any other RSU or TA to check the validity of aggregate signature by taking *params* aggregate set of $(V_i)_{i=1 \text{ to } n}$ with $(PID_i)_{i=1 \text{ to } n}$ and respective $(vpk_{PID_i})_{i=1 \text{ to } n}$ and σ_{agg} on $(m_i)_{i=1 \text{ to } n}$. It outputs true if the signature is valid or $\perp$ otherwise.

D. Security Model

In the VANETS system, to protect the shared information among the vehicles, we consider various security parameters, such as message authentication, integrity, nonrepudiation, unforgeability, unlinkability, traceability, anonymity, and resistance to impersonation, and replay and modification attacks [2], [3], [7]. As described in [12], [28], and [30], depending on the adversary behavior, we consider the following types of adversaries.

1) *Type I Adversary (Public-Key Replacement Attack):* The adversary can compromise the vehicle's secret value or capable to replace the public key of any vehicle with a value of his choice but cannot access KGC's master secret key.
2) *Type II Adversary (Malicious KGC Attack):* The adversary can access the master secret key of KGC but cannot replace the public key of any vehicle.

The existential unforgeability of a CLAS scheme can be defined by considering the following two games game-I and game-II against type-I and type-II adversaries.

Game-I: This game is executed between the challenger ξ and an adversary as follows.

1) *Initialization Phase:* In this phase, challenger ξ runs the set up algorithm to get *params*, s, and master public key. ξ then gives *params* and the master public key to $\mathcal{A}dv_1$ by keeping s secret.
2) *Queries Phase:* In this phase, $\mathcal{A}dv_1$ makes queries on the following oracles.

Reveal Partial Secret Key Oracle: On receiving a query from $\mathcal{A}dv_1$, the challenger ξ computes psk_{PID_i} by taking PID_i as input and gives this to $\mathcal{A}dv_1$.

Create User Oracle: On receiving a query from $\mathcal{A}dv_1$, the challenger ξ computes vpk_{PID_i} by taking PID_i as input and gives this to $\mathcal{A}dv_1$.

Reveal Secret Key Oracle: After receiving a query from $\mathcal{A}dv_1$, the challenger ξ returns vsk_{PID_i} by taking PID_i input.

Replace Public-Key Oracle: $\mathcal{A}dv_1$ may replace current vpk_{PID_i} with the required $vpk_{PID_i}^{\prime}$ by giving PID_i and $vpk_{PID_i}^{\prime}$.

Sign Oracle: On receiving a query from adversary $\mathcal{A}dv_1$, signing oracle returns a valid signature σ signed by current public/private key of the user PID_i, by taking PID_i, vpk_{PID_i} with message $m \in \{0,1\}^*$ as input.

1) *Forgery Phase:* Finally, $\mathcal{A}dv_1$ outputs σ_{agg}^* as forgery on messages $(m_i^*)_{i=1 \text{ to } n}$, under the identities $(PID_i^*)_{i=1 \text{ to } n}$ and the corresponding $(vpk_{PID_i}^*)_{i=1 \text{ to } n}$ and wins the game if: a) σ_{agg}^* is a valid signature;

b) partial secret key oracle and the secret key oracle have never involved in this game for at least one of $(PID_i^*)_{i=1 \text{ to } n}$, say (PID_1^*); and c) sign oracle has never been involved in this game for (PID_1^*, m_1^*).

Game-II: This game is executed between the challenger ξ and an adversary $\mathcal{A}dv_2$ as follows.

1) *Initialization Phase:* In this phase, challenger ξ runs the system initialization algorithm to get *params*, s, and master public key. The challenger then gives *params*, s, and master public key to $\mathcal{A}dv_2$.
2) *Queries Phase:* In this phase, $\mathcal{A}dv_2$ makes the following queries.

Create User Oracle: On receiving a query from $\mathcal{A}dv_2$, the challenger ξ computes vpk_{PID_i} by taking PID_i as input and gives this to $\mathcal{A}dv_2$.

Reveal Secret Key Oracle: On receiving a query from $\mathcal{A}dv_2$, the challenger ξ returns vsk_{PID_i} by taking PID_i as input.

Signing Oracle: On receiving a query from adversary $\mathcal{A}dv_2$, signing oracle returns a valid signature σ signed by current public/private key of the user PID_i, by taking PID_i, vpk_{PID_i} with message $m \in \{0, 1\}^*$ as input.

1) *Forgery Phase:* Finally, $\mathcal{A}dv_2$ outputs σ_{agg}^* as forgery on message $(m_i^*)_{i=1 \text{ to } n}$, under the identities $(PID_i^*)_{i=1 \text{ to } n}$ and the corresponding $(vpk_{PID_i}^*)_{i=1 \text{ to } n}$ and wins the game if: a) σ_{agg}^* is a valid signature; b) secret key oracle has never involved in this game for at least one of $(PID_i^*)_{i=1 \text{ to } n}$, say (PID_1^*); and c) sign oracle has never involved in this game for (PID_1^*, m_1^*).

Definition 1: A CLAS scheme is said to be existentially unforgeable under an adaptive chosen message attack, if there exists no polynomial-time adversary (type-I and type-II) with a nonnegligible advantage in the above games I and II, respectively.

IV. Proposed Authentication Scheme

In this section, we presented a novel and efficient CLAS scheme along with its security proof, under the assumption that ECDLP is intractable.

A. Proposed CLAS Scheme

The proposed CLAS scheme consists of the following nine algorithms.

1) System Setup: This algorithm generates the system necessary parameters, under the control of TA and KGC, as follows.

1) For a given security parameter λ, the TA and KGC agrees on two large primes p and q and generates an elliptic curve $E : y^2 = x^3+ax+b \bmod p$, where $a, b \in Z_p^*$ and $(4a^3 + 27b^2) \bmod p \neq 0$.
2) KGC selects $s \in Z_p^*$ at random as its master secret key and computes $P_{\text{pub}} = sP$ as the corresponding master public key. TA selects $b \in Z_p^*$ at random as its master secret key for tracking of vehicle identity and outputs $T_{\text{pub}} = bP$. Here, s is known only to KGC and b is known only to TA. Here, TA and KGC are two independent trust authorities and do not collude with each other.
3) KGC and TA chooses $H_1, H_2, H_3 : \{0, 1\}^* \rightarrow Z_p^*$ hash functions and publishes the system parameters as params $= \{P, p, q, E, G, H_1, H_2, H_3, P_{\text{pub}}, T_{\text{pub}}\}$. Any vehicle V_i sends RID_i to TA's for registration, and the *params* are stored in OBU of V_i. At the same time, RSU is also registered during the initialization phase and obtained the *params* secretly.

2) Pseudonym Identity Generation: TA generates the pseudonym identity of vehicles. The vehicle's message is communicated in a pseudonym manner to protect the vehicle's real identity information.

1) V_i selects a random number $t_i \in Z_p^*$ and computes $PID_{i,1} = t_iP, K_i = t_iT_{\text{pub}} \oplus RID_i$, and sends $\{PID_{i,1}, K_i\}$ to TA.
2) TA computes $RIDi = K_i \oplus bPID_{i,1}$ and verifies the identity. If the verification fails, it will be discarded. Otherwise, TA computes $PID_{i,2} = RID_i \oplus H_1(bP_{IDi,1}, \Delta T_i)$ and sends $PID_i = \{PID_{i,1}, PID_{i,2}, \Delta T_i\}$ to KGC secretly.

3) Partial Private Key Generation: When a vehicle V_i requests for partial private key, KGC choose $r_i \in Z_p^*$ and computes $R_i = r_iP$ and $psk_{PIDi} = (r_i + sh_{1i}) \bmod p$, where $h_{1i} = H_1(PID_i, R_i, P_{\text{pub}})$. KGC sends $\{psk_{PID_i}, R_i, PID_i\}$ to V_i and save it in its OBU. V_i can validate it $\{psk_{PID_i}, R_i, PID_i\}$ by verifying $psk_{PID_i}P = R_i + h_{1i}P_{\text{pub}}$.

4) Set Secret Value: The vehicle V_i selects $vsk_{PID_i} \in Z_p^*$ as its secret value and computes $X_i = vsk_{PID_i}P$.

5) Vehicle Key Generation: V_i generates its public key as follows: V_i outputs $h_{2i} = H_2(PID_i, X_i)$ and $Q_i = R_i + h_{2i}X_i$. V_i sets its public key as $vpk_{PIDi} = (Q_i, R_i)$ and private key as $VSK_{PIDi} = (psk_{PIDi}, vsk_{PID_i})$.

6) Signature Generation: To ensure authentication and message integrity, each message $M_i \in \{0, 1\}^*$ must be signed by a vehicle V_i. A vehicle V_i uses the current timestamp T_i and its pseudoidentity PID_i, secret value vsk_{PID_i}, and partial private key psk_{PIDi} to produce the signature as follows.

The vehicle V_i should do the following.

V_i selects $u_i \in Z_p^*$ and computes $U_i = u_iP$. V_i also computes

$$h_{2i} = H_2(PID_i, X_i)$$
$$h_{3i} = H_3(PID_i, m_i, vpk_{PIDi}, U_i, T_i)$$
$$S_i = \left[u_i + h_{3i}\left(psk_{PIDi} + h_{2i}vsk_{PID_i}\right)\right] \bmod p, \sigma_i = (U_i, S_i).$$

V_i sends $\{PIDi, vpk_{PIDi}, m_i, T_i, \sigma_i\}$ to the nearly RSU_j.

7) Signature Verification: After receiving the information $\{PIDi, vpk_{PIDi}, m_i, \Delta T_i, \sigma_i\}$ from RSU_j first verifies ΔT_i of PID_i valid and T_i is in the valid time period, then computes $h_{1i} = H_1(PID_i, R_i, P_{\text{pub}})$ and $h_{3i} = H_3(PID_i, m_i, vpk_{PIDi}, U_i, T_i)$. Checks $S_iP = U_i + h_{3i}(Q_i + h_{1i}P_{\text{pub}})$.

8) Aggregate Signature Generation: When RSU_j receives $\{PID_i, vpk_{PIDi}, m_i, T_i, \sigma_i\}$ from different vehicles V_i for $i \in \{1, 2, \ldots, n\}$ with message signature pairs $\{m_i, \sigma_i\}$, the RSU_j aggregates the multiple signatures by computing $S = \sum_{i=1}^{n} S_i$ and outputs $\sigma_{agg} = (U_1, U_2, \ldots, U_n, S)$ as CLAS.

9) Aggregate Signature Verification: After receiving $\sigma_{agg} = (U_1, U_2, \ldots, U_n, S)$ from RSU_j signed by n vehicles $V_i, i = 1, 2, \ldots, n$, with PIDs $PID_i, i = 1, 2, \ldots, n$, and the corresponding public key $vpk_{PIDi}, i = 1, 2, \ldots, n$, on $m_i, i = 1, 2, \ldots, n$, AS checks the validity of ΔT_i for PID_i of each message, and if T_i is valid for a valid interval, then AS/RSU computes hash values and check whether $SP = \sum U_i + \sum h_{3i}(Q_i + h_{1i}P_{\text{pub}})$ holds or not, where

$$h_{1i} = H_1\big(PID_i, R_i, P_{\text{pub}}\big),\;\; h_{3i} = H_3(PID_i, m_i, vpk_{PIDi}, U_i, T_i).$$

If it holds, then the aggregate signature verification is passed and these messages are accepted.

The flow of these algorithms is presented in Fig. 3.

B. Proof of Exactness

The correctness of the single signature scheme

$$\begin{aligned} S_iP &= \big(u_i + h_{3i}\big(psk_{PID_i} + h_{2i}vsk_{PID_i}\big)\big)P \\ &= U_i + h_{3i}\big(R_i + h_{1i}P_{\text{pub}} + h_{2i}X_i\big) \\ &= U_i + h_{3i}\big(Q_i + h_{1i}P_{\text{pub}}\big). \end{aligned}$$

C. Proof of Exactness of the Aggregate Signature

The correctness of the aggregate verification

$$\begin{aligned} SP &= \sum_{i=1}^{n} S_iP = \sum_{i=1}^{n}\big(U_i + h_{3i}\big(Q_i + h_{1i}P_{\text{pub}}\big)\big) \\ &= \sum_{i=1}^{n} U_i + \sum_{i=1}^{n} h_{3i}\big(Q_i + h_{1i}P_{\text{pub}}\big). \end{aligned}$$

D. Security Analysis

In this section, we present the security of the proposed CLAS scheme with respect to type I and type II adversaries. Also, we discuss the other security requirements, such as authentication, integrity, and nonrepudiation of our proposed scheme.

Theorem 1: The proposed PF-CLAS scheme is existentially unforgeable under the adaptive chosen message and identity attacks against the type-I adversary Adv_1 in the ROM provided the ECDLP is intractable by any polynomial-time-bounded algorithm in the elliptic curve group.

Proof: Let ξ be an ECDLP challenger. Let Adv_1 be a type-I polynomial-time-bounded adversary who can forge a valid aggregate signature on a message by interacting with ξ by following game-I. Now, we construct an algorithm ξ that can solve ECDLP using Adv_1. We assume that the challenger ξ is given $(P, Q = sP)$ as a random instance of ECDLP in G. Hence, ξ's goal is to find s after interacting with a type-I adversary Adv_1. For this, ξ takes PID^* as a target identity of Adv_1 on a message m^*.

1) *Initialization Phase:* Algorithm ξ sets $P_{\text{pub}} = Q = sP$ and runs the setup algorithm to generate *params* $= \{q, G, P, P_{\text{pub}}, H_i\}$ for $i = 1, 2, 3$, and master public key. ξ sends these params to $Adv_1$and by keeping s secretly.
2) *Queries Phase:* In this phase, Adv_1 asks a series of queries and these are answered by ξ adaptively.

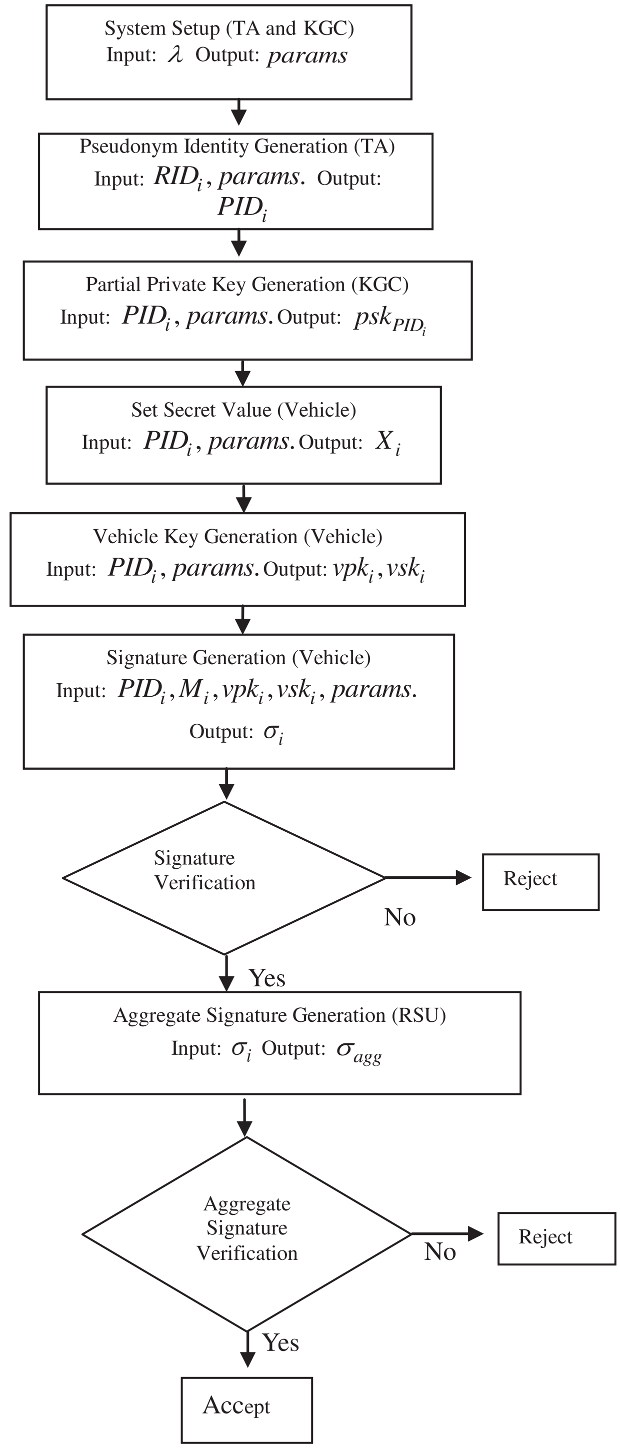

Fig. 3. Flowchart of the proposed authentication scheme.

1) *Queries on Oracle H_1 [$H_1(PID_i, R_i, P_{\text{pub}})$]:* ξ maintains an initially empty list L_1, which contains the tuple of the form $(PID_i, R_i, P_{\text{pub}}, h_{1i})$. When Adv_1 asks a query on $H_1(PID_i, R_i, P_{\text{pub}})$, ξ returns h_{1i} if such tuple exists in L_1. If not, ξ selects a random $h_{1i} \in Z_q^*$ and sets $H_1(PID_i, R_i, P_{\text{pub}}) = h_{1i}$. ξ then returns h_{1i} to Adv_1 and inserts the tuple $(PID_i, R_i, P_{\text{pub}}, h_{1i})$ to the list L_1.
2) *Queries on Oracle H_2 [$H_2(PID_i, X_i)$]:* ξ maintains an initially empty list L_2, which contains the tuples of the form (PID_i, X_i, h_{2i}). When Adv_1 asks a H_2 query on (PID_i, X_i), ξ returns h_{2i} if such tuple already exists

in L_2. If not, ξ selects a random $h_{2i} \in Z_q^*$ and sets $H_2(PID_i, X_i) = h_{2i}$. ξ then returns h_{2i} to Adv_1 and inserts the tuple (PID_i, X_i, h_{2i}) to the list L_2.

3) *Queries on Oracle H_3 [$H_3(PID_i, m_i, vpk_i, U_i)$]:* ξ maintains an initially empty list L_3, which contains the tuples of the form $(PID_i, m_i.vpk_i, U_i, h_{3i})$. When Adv_1 asks a H_3 query on (PID_i, m_i, vpk_i, U_i), ξ returns h_{3i} if such tuple already exists in L_3. If not, ξ selects a random $h_{3i} \in Z_q^*$ and sets $H_3(PID_i, m_i, vpk_i, U_i) = h_{3i}$. ξ then returns h_{3i} to Adv_1 and add the tuple $(PID_i, m_i, vpk_i, U_i, h_{3i})$ to the list L_3.

4) *Reveal Partial Secret Key Oracle PSK(PID_i):* ξ maintains an initially empty list L_{psk} that contains the tuples of the form (PID_i, R_i, d_i). When Adv_1 asks a query on $PSK(PID_i)$, ξ returns $D_i = (d_i, R_i)$, if such tuple already exists in L_{psk}. Otherwise, ξ does as follows. If $PID_i = PID^*$, ξ aborts. If $PID_i \neq PID^*$, ξ chooses $a_i, b_i \in Z_q^*$ and sets $d_i = a_i$, $H_1(PID_i, R_i, P_{\text{pub}}) = b_i$ and $R_i = a_iP - b_iP_{\text{pub}}$. Now, ξ adds $(PID_i, R_i, P_{\text{pub}}, b_i)$ to L_1 list and adds the tuple (PID_i, R_i, d_i) to L_{psk} list.

5) *Create User Oracle Cuser(PID_i):* ξ maintains an initially empty list L_{Cuser} that contains the tuple of the form $(PID_i, Q_i, R_i, x_i, d_i)$. When Adv_1 asks a query on Cuser(PID_i), ξ returns the current public key $vpk_i = (Q_i, R_i)$, if such tuple already exists in L_{Cuser}. Otherwise, ξ does the following: If $PID_i = PID^*$, ξ chooses $a_i, b_i, c_i, x_i \in Z_q^*$ and sets $R_i = a_iP$, $H_1(PID_i, R_i, P_{\text{pub}}) = b_i, X_i = x_iP$, and $H_2(PID_i, X_i) = c_i$. Now, ξ sets $Q_i = a_iP + c_i(x_iP)$. Now, ξ adds $(PID_i, R_i, P_{\text{pub}}, b_i)$ to the list L_1 and (PID_i, X_i, c_i) to L_2 and $(PID_i, Q_i, R_i, x_i, \perp)$ to L_{Cuser}. Now, ξ returns the public key $vpk_i = (Q_i, R_i)$ to Adv_1. If $PID_i \neq PID^*$, ξ recovers the tuple (PID_i, R_i, d_i) from L_{psk} and chooses $c_i, x_i \in Z_q^*$ and sets $X_i = x_iP$ and $H_2(PID_i, X_i) = c_i$. Now, ξ sets $Q_i = R_i + c_iX_i = R_i + h_{2i}X_i$ and outputs $vpk_i = (Q_i, R_i)$ as public key. Here, ξ adds (PID_i, X_i, c_i) to L_2 and $(PID_i, Q_i, R_i, x_i, d_i)$ to L_{Cuser} list.

6) *Reveal Secret Value Oracle RSK(ID_i):* When Adv_1 asks a query on $RSK(ID_i)$, ξ does as follows. If $PID_i = PID^*$, ξ aborts the simulation. If $PID_i \neq PID^*$, ξ recovers the tuple $(PID_i, Q_i, R_i, x_i, d_i)$ from L_{Cuser} and sends x_i to Adv_1. If such tuple does not exist in L_{Cuser}, then ξ makes a query on Cuser(PID_i) to produce (x_i, Q_i) and adds this value to the list L_{Cuser}. Finally, ξ returns x_i as a secret value.

7) *Replace Public-Key Oracle RPK(PID_i):* If Adv_1 wants to replace the public key $vpk_i = (Q_i, R_i)$ of PID_i with a value of his choice $vpk_i' = (Q_i', R_i')$, then ξ finds the tuple $(PID_i, Q_i, R_i, x_i, d_i)$ from the list L_{Cuser} and then updates Q_i with Q_i' and R_i with R_i'. Now, ξ sets $x_i' = \perp$ and $d_i = \perp$. Hence, the replaced tuple is of the form $(PID_i, Q_i', R_i', \perp, \perp)$.

8) *Signing Oracle:* When Adv_1 asks a query on (PID_i, m_i), ξ does as follows. If $PID_i \neq PID^*$, then ξ recovers the corresponding tuple, such as $(PID_i, R_i, P_{\text{pub}}, h_{1i}), (PID_i, X_i, h_{2i})$, and $(PID_i, Q_i, R_i, x_i, d_i)$ from L_1, L_2, and L_{Cuser} lists, respectively, and generates a valid signature as follows. ξ choose $u_i, h_{3i} \in Z_q^*$ and compute $U_i = u_iP, v_i = u_i + h_{3i}(d_i + h_{2i}x_i) \bmod q$. Now, ξ returns $\sigma_i = (U_i, v_i)$ to Adv_1 as a valid signature and adds $(PID_i, m_i, vpk_i, U_i, h_{3i})$ to L_3. If $PID_i = PID^*$, then ξ recovers $(PID_i, R_i, P_{\text{pub}}, h_{1i})$ from the list L_1 and $(PID_i, Q_i, R_i, x_i, d_i)$ from L_{cuser} lists. Here, $x_i' = \perp$ and $d_i = \perp$. Now, ξ chooses $u_i, h_{3i} \in Z_q^*$ and sets $v_i = u_i$ and $U_i = v_iP - h_{3i}(Q_i + h_{1i}P_{\text{pub}})$. ξ returns $\sigma_i = (U_i, v_i)$ as a valid signature to Adv_1 and adds $(PID_i, m_i, vpk_i, U_i, h_{3i})$ to the list L_3. Now, Adv_1 can compute the final aggregate signature $\sigma_{agg} = (U_i, S)$ for $i = 1$ to n from the individual signatures $\sigma_i = (U_i, S_i)$ for $i = 1$ to n.

Forgery/Output: Finally, Adv_1 returns a set of n vehicles $(V_i)_{i=1 \text{ to } n}$, whose identities $(PID_i)_{i=1 \text{ to } n}$ and corresponding public keys $(vpk_i)_{i=1 \text{ to } n}$ with n messages $(m_i)_{i=1 \text{ to } n}$, the timestamp T_i and a forged aggregate signature $\sigma_{agg}^* = (U_i^*, S^*)_{i=1 \text{ to } n}$, i.e., $(U_1^*, U_2^*, \ldots, U_n^*, S^*)$. If $PID_i \neq PID^*$, ξ stops the simulation, otherwise, ξ does the following. Let $\sigma_{agg}^* = (U_i^*, S^*)_{i=1 \text{ to } n}$ denote as $\sigma_{agg}^{*(1)} = (U_i^*, S^{*(1)})_{i=1 \text{ to } n}$, by using the Forking lemma [45], after replying ξ with the same random string but different hash function H_3, the Adv_1 can obtain another $2n$ convincing aggregate signatures as $\sigma_{agg}^{*(j)} = (U_i^*, S^{*(j)})_{i=1 \text{ to } n}, j = 2, 3, \ldots, 2n + 1$. Since $\sigma_{agg}^{*(j)}$ satisfy the aggregate verification equation, thus we have $S^{*(j)}P = \sum_{i=1}^{n} U_i^* + \sum_{i=1}^{n} h_{3i}^{*(j)}(Q_i^* + h_{1i}^*P_{\text{pub}}), j = 1, 2, 3, \ldots, (2n+1)$. By u_i^*, q_i^*, and s, we now denote the discrete logarithms of U_i^*, Q_i^*, and P_{pub}, respectively, i.e., $U_i^* = u_i^*P, Q_i^* = q_i^*P$ for $i = 1$ to n, and $P_{\text{pub}} = sP$. From these $(2n + 1)$ equations, we could get the following $(2n + 1)$ equations. $S^{*(j)} = \sum_{i=1}^{n} u_i^* + \sum_{i=1}^{n} h_{3i}^{*(j)}(q_i^* + h_{1i}^*s), j = 1, 2, 3, \ldots, (2n+1)$. In these equations u_i^*, q_i^* for $i = 1$ to n and s are unknown to ξ. Now, ξ can solve these values from the above $(2n + 1)$ linearly independent equations, and output s as the solution of ECDLP.

Finally, $\xi's$ success probability in solving the ECDLP is at least $[1/(q_{psk} + n)](1 - [1/(q_{psk} + n)])^{(q_{psk}+n-1)} \varepsilon$, and for large $q_{psk,}$ this probability turns to $[1/((q_{psk} + n)e)]\varepsilon$. Hence, given an instance $(P, Q = sP)$, ξ can solve ECDLP with nonnegligible probability $[1/((q_{psk} + n)e)]\varepsilon$, where q_{psk} is the number of queries on *reveal partial secret key oracle,* n is the number of aggregate signers, and e is the base of the natural logarithm, which is a contradiction with ECDLP assumption. ■

Theorem 2: The proposed PF-CLAS scheme is existentially unforgeable under the adaptive chosen message and identity attacks against the type-II adversary Adv_2 in ROM provided the ECDLP is intractable by any polynomial-time-bounded algorithm in the elliptic curve group.

Proof: Let ξ be an ECDLP challenger. Let Adv_2 be a type-II polynomial-time-bounded adversary who can forge a valid signature on a message by interacting with ξ by following game-II. Now, we construct an algorithm ξ that can solve ECDLP using Adv_2. Challenger ξ is given $(P, Q = \alpha P)$ as a random instance of ECDLP in G. Hence, ξ's goal is to find "α" after interacting with type-II adversary Adv_2. For this, ξ takes PID^* as a target identity of Adv_2 on a message m^*.

1) *Initialization Phase:* Challenger ξ picks a random $s \in Z_q^*$ and runs the setup algorithm by setting $P_{\text{pub}} = sP$ and generates the system parameters as params $= \{q, G, P, P_{\text{pub}}, H_i$ for $i = 1, 2, 3\}$. ξ gives *params* and master secret key to Adv_2.
2) *Queries Phase:* In this phase, Adv_2 asks a series of queries, and these are answered by ξ adaptively.
1) *Queries on Oracle H_1 [$H_1(PID_i, R_i, P_{\text{pub}})$]:* ξ maintains an initially empty list L_1, which contains the tuple of the form $(PID_i, R_i, P_{\text{pub}}, h_{1i})$. When Adv_2 asks a query on $H_1(PID_i, R_i, P_{\text{pub}})$, ξ returns h_{1i} if such tuple exists in L_1. If not, ξ selects a random $h_{1i} \in Z_q^*$ and inserts to the list L_1. At last, ξ gives h_{1i}.
2) *Queries on Oracle H_2 [$H_2(PID_i, X_i)$]:* ξ maintains an initially empty list L_2, which contains the tuples of the form (PID_i, X_i, h_{2i}). When Adv_2 asks a query on (PID_i, X_i), ξ returns h_{2i} if such tuple already exists in L_2. If not, ξ selects a random $h_{2i} \in Z_q^*$ and inserts to the list L_2. Finally, ξ gives h_{2i}.
3) *Queries on Oracle H_3 [$H_3(PID_i, m_i, vpk_i, U_i)$]:* ξ maintains an initially empty list L_2, which contains the tuples of the form $(PID_i, m_i, vpk_i, U_i, h_{3i})$. When Adv_2 asks a query on (PID_i, m_i, vpk_i, U_i), ξ returns h_{3i} if such tuple already exists in L_3. If not, ξ selects a random $h_{3i} \in Z_q^*$ and adds to the list L_3. Finally, ξ gives h_{3i}.
4) *Create User Oracle* Cuser*(ID_i):* ξ maintains an initially empty list L_{Cuser} that contains the tuple of the form $(PID_i, Q_i, R_i, x_i, d_i)$. When Adv_2 asks a query on Cuser(ID_i), ξ returns the current public key $vpk_i = (Q_i, R_i)$, if such tuple already exists in L_{Cuser}. Otherwise, ξ does as follows.
   a) If $PID_i = PID^*$, ξ chooses $a_i, b_i, c_i \in Z_q^*$ and sets $a_i, b_i, c_i, x_i \in Z_q^* R_i = a_iP, H_1(PID_i, R_i, P_{\text{pub}}) = b_i$ and $X_i = Q = \alpha P$ and $H_2(PID_i, X_i) = c_i$. Now, ξ sets $Q_i = R_i + h_{2i}X_i = a_iP + c_i(\alpha P)$. Now ξ adds $(PID_i, R_i, P_{\text{pub}}, b_i)$ to L_1 and (PID_i, X_i, c_i) to L_2 and $(PID_i, Q_i, R_i, \perp, d_i)$ to L_{Cuser}. Now, ξ returns the public key $vpk_i = (Q_i, R_i)$ to Adv_2.
   b) If $PID_i \neq PID^*$, ξ chooses $a_i, b_i, c_i, x_i \in Z_q^*$ and sets $R_i = a_iP, H_1(PID_i, R_i, P_{\text{pub}}) = b_iX_i = x_iP$, and $H_2(PID_i, X_i) = c_i$. Now, ξ sets $Q_i = R_i + h_{2i}X_i = a_iP + c_i(x_iP)$. Now, ξ adds $(PID_i, R_i, P_{\text{pub}}, b_i)$ to L_1 and (PID_i, X_i, c_i) to L_2 and $(PID_i, Q_i, R_i, x_i, d_i)$ to L_{Cuser}. Now, ξ returns the public key $vpk_i = (Q_i, R_i)$ to Adv_2.
5) *Reveal Secret Value Oracle $RSK(PID_i)$*: When Adv_2 asks a query on $RSK(PID_i)$, ξ does as follows.
   a) If $PID_i = PID^*$, ξ aborts the simulation.
   b) If $PID_i \neq PID^*$, ξ recovers the tuple $(PID_i, Q_i, R_i, x_i, d_i)$ from L_{Cuser} and sends x_i to Adv_2. If such tuple does not exists in L_{Cuser} list, then ξ makes a query on Cuser(PID_i) to produce (x_i, Q_i) and adds this to the list L_{Cuser}. Finally, ξ returns x_i as a secret value.
6) *Signing Oracle:* When Adv_2 asks a query on (PID_i, m_i), ξ does as follows.
   a) If $PID_i \neq PID^*$, then ξ recovers the corresponding tuple, such as (PID_i, X_i, h_{2i}) and $(PID_i, Q_i, R_i, x_i, d_i)$ from L_2 and L_{Cuser} lists, respectively, and generates a valid signature as follows. Choose $u_i, h_{3i} \in Z_q^*$ and set $U_i = u_iP$ and compute $v_i = u_i + h_{3i}(d_i + h_{2i}x_i) \bmod q$. Now, ξ returns $\sigma_i = (U_i, v_i)$ to Adv_2 as a valid signature and adds $(PID_i, m_i, vpk_i, U_i, h_{3i})$ to L_3.
   b) If $PID_i = PID^*$, then ξ recovers corresponding tuples from L_2 and L_{Cuser} lists, respectively, i.e., (PID_i, X_i, h_{2i}) and $(PID_i, Q_i, R_i, \perp, d_i)$ tuples and generates the signature as follows: ξ chooses $u_i, h_{3i} \in Z_q^*$and sets $U_i = h_{3i}(u_iP - h_{2i}X_i)$ and $v_i = h_{3i}(d_i + u_i)$. Now, ξ returns $\sigma_i = (U_i, v_i)$ to Adv_2 as a valid signature and adds $(PID_i, m_i, vpk_i, U_i, h_{3i})$ to L_3. The signature generated in this way is valid.

Now, the adversary Adv_2 can compute the final aggregate signature $\sigma_{agg} = (U_i, S)$ for $i = 1$ to n from the individual signatures $\sigma_i = (U_i, S_i)$ for $i = 1$ to n.

Forgery/Output: Finally, Adv_1 returns a set of n vehicles $(V_i)_{i=1 \text{ to } n}$, whose identities $(PID_i)_{i=1 \text{ to } n}$ and corresponding public keys $(vpk_i)_{i=1 \text{ to } n}$ with n messages $(m_i)_{i=1 \text{ to } n}$, the timestamp T_i and a forged aggregate signature $\sigma_{agg}^* = (U_i^*, S^*)_{i=1 \text{ to } n}$, i.e., $(U_1^*, U_2^*, \ldots, U_n^*, S^*)$.

If $PID_i \neq PID^*$, ξ stops the simulation, otherwise, ξ does the following. Let $\sigma_{agg}^* = (U_i^*, S^*)_{i=1 \text{ to } n}$ denote as $\sigma_{agg}^{*(1)} = (U_i^*, S^{*(1)})_{i=1 \text{ to } n}$, by using the Forking lemma [45], after replaying ξ with the same random tape but different choice of H_3, the Adv_2 can obtain another n convincing aggregate signatures as $\sigma_{agg}^{*(j)} = (U_i^*, S^{*(j)})_{i=1 \text{ to } n}, j = 2, 3, \ldots, n+1$. Since $\sigma_{agg}^{*(j)}$ satisfies $S^{*(j)}P = \sum_{i=1}^n U_i^* + \sum_{i=1}^n h_{3i}^{*(j)}(Q_i^* + h_{1i}^* P_{\text{pub}}), j = 1, 2, 3, \ldots, n+1$, or $S^{*(j)}P = \sum_{i=1}^n U_i^* + \sum_{i=1}^n h_{3i}^{*(j)}(d_i^* P + h_{2i}^* X_i^*)$. The values u_i^* and α, are the discrete logarithms of U_i^* and X_i^*, respectively. Here, $X_i^* = \alpha P$ for $i = 1$ to n and d_i^* is known to ξ, i.e., $U_i^* = u_i^* P$ and $X_i^* = \alpha P$ for $i = 1$ to n. From the above $(n+1)$ equations, we can get the following $(n+1)$ equations, $S^{*(j)} = \sum_{i=1}^n u_i^* + \sum_{i=1}^n h_{3i}^{*(j)}(d_i^* + h_{2i}^* \alpha), j = 1, 2, 3, \ldots, n+1$.

In these equations, u_i^* and α are unknown values and $S^{*(j)}$ and d_i^* are known values to ξ. Now, ξ can solve these values from the above $(n+1)$ linearly independent equations and output α as the solution of ECDLP. ■

E. Other Security Requirements

The proposed CLAS scheme achieves the following security requirements for secure communication between vehicles.

1) *Message Authentication and Integrity:* These two properties can be achieved directly from the unforgeability proof of Theorems 1 and 2.
2) *Nonrepudiation:* Since TA can relate the real identity and pseudoidentity of a message so that no vehicle can deny its signature on a message.
3) *Unlinkability:* In our scheme, the verifier V_i sends $\{PIDi, vpk_{PIDi}, m_i, T_i, \sigma_i\}$ to the nearby RSU. No attacker can relate two messages of the same vehicle due to randomness of u_i in the signature of σ_i. So the proposed scheme satisfies unlinkability property.

4) *Traceability:* The use of pseudoidentity does not allow an adversary to trace the trajectory of the vehicle. However, in certain circumstances (for, e.g., accidents and traffic jams), the real identity of the vehicle should be retrieved by vehicle authorities. This conditional traceability enable the TA to recover the real identity of the vehicle from its pseudoidentity $RID_i = K_i \oplus bPID_{i,1}$. Hence, only TA is able to trace the real identity of any malicious vehicle.
5) *Anonymity:* The vehicles' real identity is kept perfectly anonymous in our scheme since the real identity of the vehicle is not known to other vehicles and RSU except TA. The use of pseudoidentity of these vehicles does not allow an adversary or other vehicles to trace the trajectory of the vehicle. Since the vehicle uses pseudoidentity, as $PID_{i,1} = t_iP$ and $PID_{i,2} = RID_i \oplus H_1(bP_{IDi,1}, \Delta T_i)$; privacy and anonymity of the vehicle can be achieved.
6) *Resistance to Various Attacks:* Due to Theorems 1 and 2, our scheme is able to resist various attacks, such as impersonation, replay, and modification attacks.

V. EFFICIENCY ANALYSIS

This section discusses the performance of our CLAS authentication scheme with respect to framework, security, aggregation type, computational complexity, and transmission overhead. For the evaluation of these parameters, we consider the experimental results from the works [46]–[49], where various cryptographic operations are evaluated using MIRACL software on Pentium IV and are listed in Table II. The operations and their conversions presented in Table II are achieved by considering the points on super singular elliptic curve $E/F_p : y^2 = x^3 + x$ built with Solinas prime ordered group with 512-b prime number p satisfying $p+1 = 12qr$. Table III presents the comparison of our scheme with existing CLAS schemes [4], [6], [28]–[30], [37], [39]–[44] in terms of under lying hard problem, security of the scheme, type of aggregation, and with/without pairings. Most of these CLAS schemes presented in Table III are insecure due to various types of attacks. Hence, we consider only secure CLAS schemes [6], [29], [37], [40], [43], [44] for comparison with our scheme.

TABLE II
NOTATION AND DESCRIPTION OF VARIOUS CRYPTOGRAPHIC OPERATION AND THEIR CONVERSIONS

Notations	Description
T_{MM}	Modular multiplication operation in Z_q^* $1T_{MM} \approx 0.2325ms$
T_{SM}	Elliptic curve point multiplication, (Scalar multiplication in G_{Adt}), $T_{SM} = 29T_{MM} \approx 6.38ms$
T_{BP}	Bilinear pairing in G_{Mlt}, $T_{BP} = 87T_{MM} \approx 20.01ms$
T_H	Simple hash function which is negligible
T_{MTPH}	Map to point hash function, $1T_{MTPH} = 1T_{SM} = 29T_{MM} \approx 6.38ms$
T_{MX}	Modular Exponentiation operation, $1T_{MX} = 240T_{MM} \approx 55.20ms$

TABLE III
COMPARISON OF THE PROPOSED SCHEME WITH RELATED SCHEMES

Scheme	Hard Problem	Pairing based /Pairing free	Type of Aggregation	Security
Malhi et al. [30]	CDHP	Pairing based	Partial	Insecure
Horng et al. [28]	CDHP	Pairing based	Partial	Insecure
Li et al. [29]	CDHP	Pairing based	Partial	Secure
Kumar et al. [6]	CDHP	Pairing based	Partial	Secure
Zhong et al. [39]	CDHP	Pairing based	Full	Insecure
Kamil et al. [40]	CDHP	Pairing based	Full	Secure
Wang et al. [37]	CDHP	Pairing based	Partial	Secure
Mei et al. [43]	CDHP	Pairing based	Full	Secure
Xu et al. [44]	CDHP	Pairing based	Partial	Secure
Cui et al. [4]	ECDLP	Pairing free	Partial	Insecure
Kamil et al. [41]	ECDLP	Pairing free	Full	Insecure
Zhao et al. [42]	ECDLP	Pairing free	Partial	Insecure
Ours	ECDLP	Pairing free	Partial	Secure

Computation Costs: In the following, we present the computational complexity of our scheme and other existing secure CLAS schemes for VANETs [6], [29], [37], [40], [43], [44]. To evaluate the computational complexity, we consider the signing cost, verification cost of individual signatures, and also aggregate signature. Li *et al.* scheme [29] requires $2T_{SM} + 1T_{PA} + 1T_{MTPH} = 87.12T_{MM}$ for signing and $3T_{BP} + 1T_{SM} + 1T_{PA} + 2T_{MTPH} = 348.12T_{MM}$ for verification. Hence, the total computational cost for single signature of Li *et al.* scheme [29] is $435.24T_{MM}$. Li *et al.* scheme [29] requires $(n+1)T_{MTP} + 3T_{BP} + nT_{SM} + (3n-2)T_{PA}$ for verification of n signatures. So the total cost for verification of aggregate signature (for $n = 100$) is $6125.76\,T_{MM}$. Since the proposed scheme is pairing free, it requires only one scalar multiplication ($1T_{SM} = 29\,T_{MM}$) for signature generation and three scalar multiplications and two point additions ($3T_{SM} + 2T_{PA} = 87.24\,T_{MM}$) for signature verification. For verification of n signatures, the proposed scheme requires $(2n+1)T_{SM} + (3n-1)T_{PA}$. So the total verification cost of our aggregate signature is $5864.88\,T_{MM}$. Similarly, we computed computational cost of the existing secure CLAS schemes [6], [29], [37], [40], [43], [44] and is presented in Table IV. The computation cost for signing and verification of these secure CLAS signature schemes are represented graphically in Fig. 4. The computation cost for n signatures versus aggregate signature was presented through a bar graph as shown in Fig. 5 for various numbers of signatures from various vehicles. From Fig. 5, it is clear that the computation cost of aggregate signature is significantly more efficient than other secure schemes. Delay in signing a message, verifying a message, and aggregate verification of messages with respect to the number of messages were presented through graphs as shown in Figs. 6–8, respectively.

Transmission Overhead: The transmission overhead of our scheme with the other CL-based aggregate signatures for VANETs [6], [29], [37], [40], [43], [44] are calculated and compared. Out of all secure schemes, our scheme is the only scheme with pairing-free environment and it is based on ECC. The schemes Kumar *et al.* [6],

TABLE IV
COMPARISON OF THE PROPOSED SCHEME WITH RELATED SCHEMES

Scheme	Signing Cost	Verification Cost	Aggregate Verification Cost (for n=100)
Kumar et al. [6]	$4T_{SM}+2T_{PA}+1T_{MTP}$ $=145.24T_{MM}\approx 33.7683ms$	$4T_{BP}+3T_{SM}+2T_{MTP}$ $=493T_{MM}\approx 114.6225ms$	$4T_{BP}+3nT_{SM}+3(n-1)T_{PA}$ $=9083.64T_{MM}\approx 2111.9463ms$
Li et al. [29]	$2T_{SM}+1T_{PA}+1T_{MTP}$ $=87.12T_{MM}\approx 20.2554ms$	$3T_{BP}+1T_{SM}+1T_{PA}+1T_{MTP}$ $=348.12T_{MM}\approx 80.9379ms$	$(n+1)T_{MTP}+3T_{BP}+nT_{SM}+(3n-2)T_{PA}$ $=6125.76T_{MM}\approx 1424.2392ms$
Wang et al. [37]	$4T_{SM}+2T_{PA}=116.24T_{MM}$ $\approx 27.0258ms$	$3T_{BP}+3T_{SM}+1T_{MTP}+1T_{PA}$ $=377.12T_{MM}\approx 87.6804ms$	$3T_{BP}+3nT_{SM}+nT_{MTP}+(3n-2)T_{PA}$ $=11896.76T_{MM}\approx 2765ms$
Kamil et al. [40]	$4T_{SM}+2T_{PA}+1T_{MTP}$ $=145.24T_{MM}\approx 33.7683ms$	$3T_{BP}+2T_{SM}+2T_{MTP}+1T_{PA}$ $=377.12T_{MM}\approx 87.6804ms$	$3T_{BP}+2nT_{SM}+(2n-1)T_{PA}$ $=6084.88T_{MM}\approx 1414.7346ms$
Mei et al. [43]	$4T_{SM}+2T_{PA}+2T_{MTP}$ $=174.24T_{MM}\approx 40.5108ms$	$4T_{BP}+2T_{SM}+2T_{MTP}$ $=464T_{MM}\approx 107.88ms$	$4T_{BP}+2nT_{SM}+2T_{MTP}+(2n-2)T_{PA}$ $=6229.76T_{MM}\approx 1448.4192ms$
Xu et al. [44]	$3T_{SM}+1T_{PA}+1T_{MTP}$ $=116.12T_{MM}\approx 26.9979ms$	$3T_{BP}+2T_{SM}+2T_{MTP}+1T_{PA}$ $=377.12T_{MM}\approx 87.68ms$	$3T_{BP}+2nT_{SM}+(n+1)T_{MTP}+(3n-2)T_{PA}$ $=9025.76T_{MM}\approx 2098.4892ms$
Our Scheme	$1T_{SM}=29T_{MM}$ $\approx 6.7425ms$	$3T_{SM}+2T_{PA}$ $=87.24T_{MM}\approx 20.2833ms$	$(2n+1)T_{SM}+(3n-1)T_{PA}$ $=5864.88T_{MM}\approx 1363.5846ms$

TABLE V
LENGTH OF THE GROUP IN BILINEAR PAIRING AND ECC

Type of the System	Type of the Curve	Pairing	Cyclic group	$\lvert p\rvert,\lvert p\rvert$	$\lvert G\rvert$	Length of elements of the group
Bilinear Pairing	$E: y^2=x^3+x \bmod p$	$\hat{e}: G_1\times G_1\rightarrow G_T$	$G_1(P)$	$\lvert p\rvert=512$ bits	$q=160$ bits	$\lvert G_1\rvert=1024$ bits
ECC	$E: y^2=x^3+ax+b \bmod p, a,b\in Z_q^*$.	Without Pairing	$G(P)$	$\lvert p\rvert=160$ bits	$q=160$ bits	$\lvert G\rvert=320$ bits

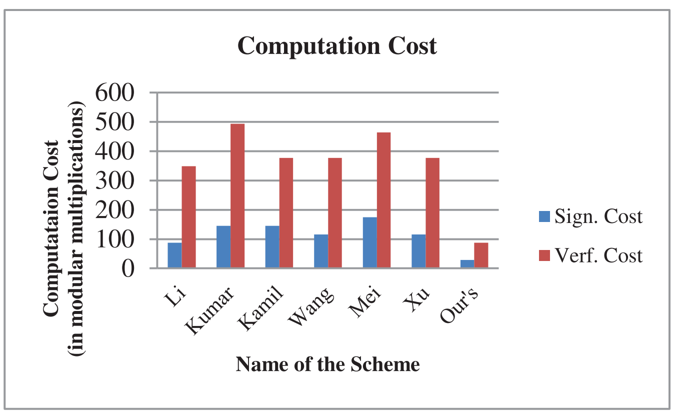

Fig. 4. Computation cost for signature generation and verification.

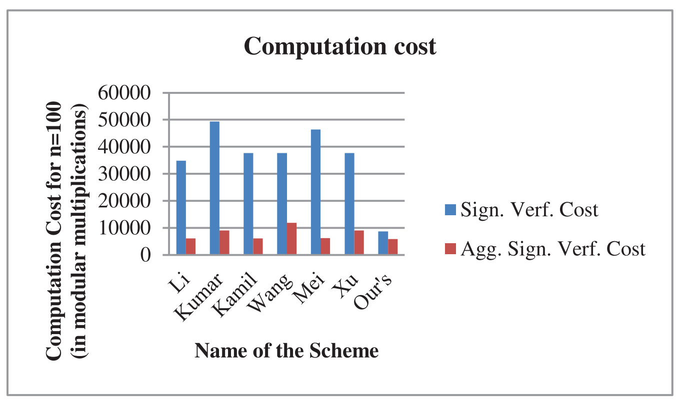

Fig. 5. Computation cost for n individual verification and aggregate signature verification cost.

Li *et al.* [29], Wang and Teng [37], Kamil and Ogundoyin [40], Mei *et al.* [43], and Xu *et al.* [44] are designed using bilinear pairings. We consider the parameters, such as curve type, order of the group, and length of the elements of the group for pairing-based schemes and ECC-based schemes as shown in Table V. These specifications provide an equivalent 1024-b RSA security level. The transmission overhead includes the length of signature, pseudoidentity, current timestamp, psk_{PIDi}, vsk_{PID_i} only but not message. In our scheme, the vehicle sends $PID_i, vpk_{PID_i}, \sigma_i = (U_i, S_i), T_i$, where $PID_i, vpk_i \in G$ and $S_i \in Z_q^*$, and T_i is current timestamp. The total transmission overhead is $4|G| + |Z_q^*| + |T_k| = 1472$ bits. Similarly, the total transmission cost for Li *et al.* schemed [29] Kumar *et al.* [6], Wang and Teng [37], Kamil and Ogundoyin [40], Mei *et al.* [43], and Xu *et al.* [44] are calculated. The comparison of total signature length and the transmission overhead for single message and for n messages was presented in Table VI. The transmission overhead of our scheme is presented in Fig. 9. The communication overhead with respect to the number of messages was presented in Fig. 10.

Power Utilization: Now, we analyze the efficiency of our scheme in terms of power utilization and compare with other secure schemes [6], [29], [37], [40], [43], [44]. Power

TABLE VI
COMPARISON OF THE COMMUNICATION AND TRANSMISSION OVERHEAD

Scheme	Transmission for single message	Transmission for n messages	Sign. Length	Sign. Lengtl (in bytes)	Aggr. Sign. Length
[6]	768 bytes	768n bytes	$2\|G_1\|$	256	$(n+1)\|G_1\|$
[29]	689 bytes	689n bytes	$2\|G_1\|$	256	$(n+1)\|G_1\|$
[37]	660 bytes	660n bytes	$2\|G_1\|$	256	$(n+1)\|G_1\|$
[40]	680 bytes	680n bytes	$2\|G_1\|$	256	$2\|G_1\|$
[43]	680 bytes	680n bytes	$2\|G_1\|$	256	$2\|G_1\|$
[44]	404 bytes	404n bytes	$2\|G_1\|$	256	$(n+1)\|G_1\|$
Our Scheme	184 bytes	184n bytes	$\|G\|+\|Z_q^*\|$	60	$n\|G\|+\|Z_q^*\|$

TABLE VII
COMPARISON OF THE POWER UTILIZATION WITH OTHER SCHEMES

Scheme	Signing Cost (in *mj*)	Verification Cost (in *mj*)	Aggregate Verification Cost (in *mj*) (for n=100)
Kumar et al. [6]	367	1247	22978
Li et al. [29]	220	881	15495
Wang et al. [37]	294	953	30094
Kamil et al. [40]	367	953	15392
Mei et al. [43]	440	1173	15758
Xu et al. [44]	293	953	22831
Our Scheme	73	221	14836

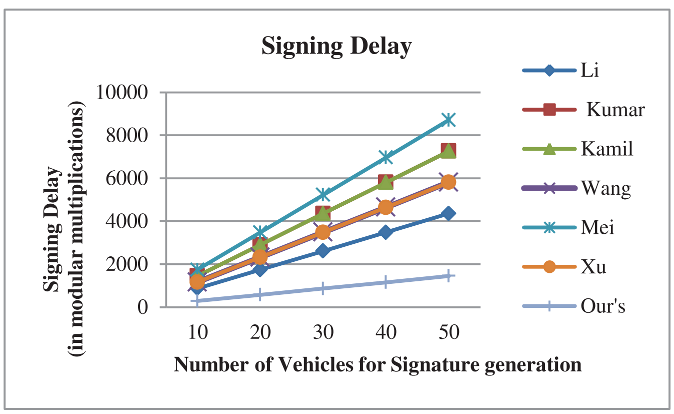

Fig. 6. Delay in signing with respect to the number of vehicles.

utilization can be calculated as $E = tP$, where t is the time taken to generate or verify a message, P is the maximum power of CPU (10.88 W), and E is the consumed power. The results of power utilization for the sign, verify, and aggregation are presented in Table VII. The power utilization comparison for signature generation and signature verification (single data verification) is shown in Fig. 11. It can be observed that the power utilized by a signer and verifier in the proposed scheme is significantly less than the existing secured schemes. The power utilization increases with the number of participants as shown in Fig. 12. But the proposed scheme requires less power than other schemes when the number of users increases in the system. As the number of participants increases, the proposed scheme consumes less power than those of the other schemes.

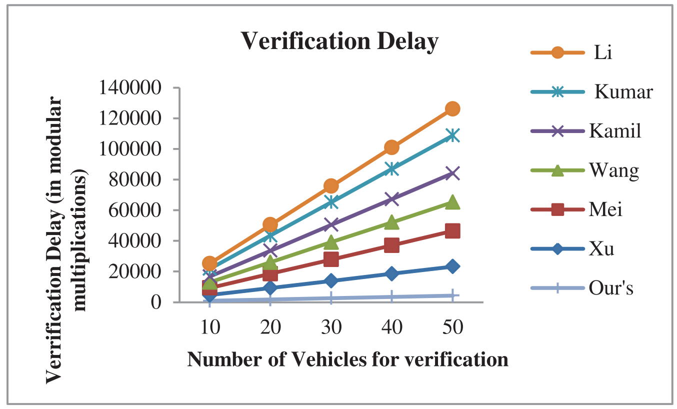

Fig. 7. Delay in verification with respect to the number of vehicles.

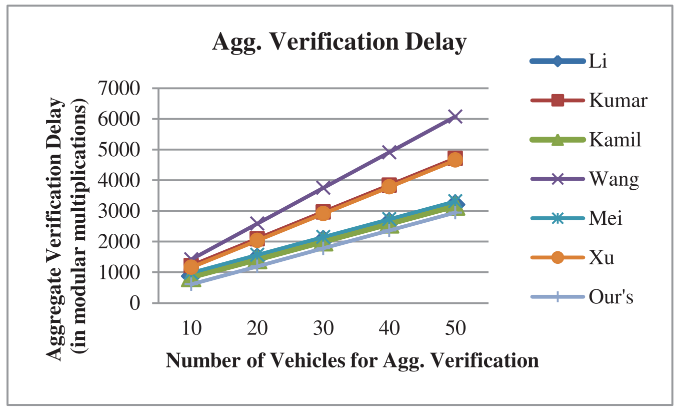

Fig. 8. Delay in aggregate verification of messages with respect to the number of vehicles.

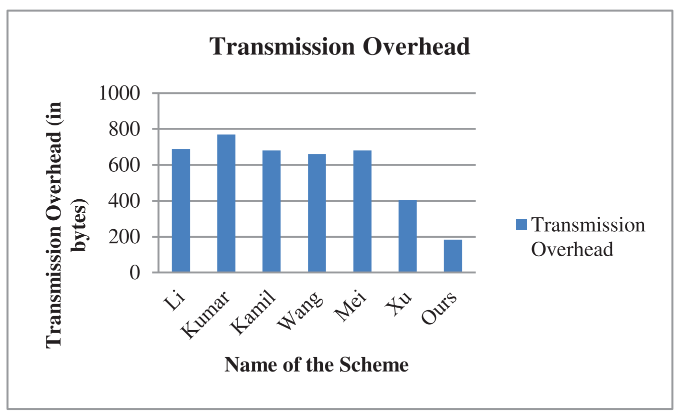

Fig. 9. Graphical presentation of transmission overhead.

Table VI presents the comparison of length of the signature our scheme and other existing secure CLAS schemes [6], [29], [37], [40], [43], [44]. In VANETS, aggregation is performed by RSU by receiving different messages from different vehicles and can be transmitted by the generated aggregate signature to other RSUs or AS or TA. In this way, RSU can reduce the communication cost with other RSUs or TA. However, as the RSU has to verify n signatures received from n different vehicles, the signature length for a single message plays a vital role in comparing with aggregate signature length. As

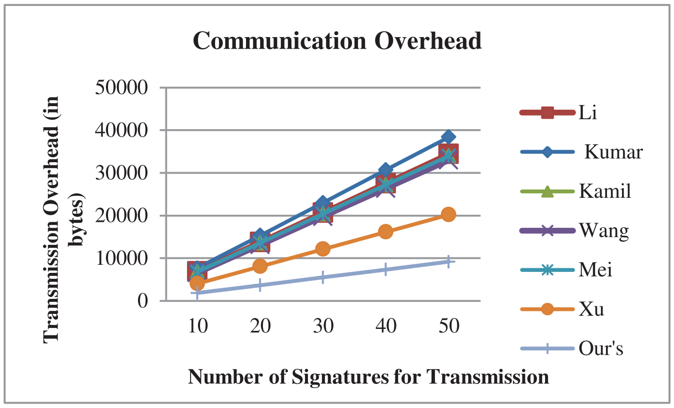

Fig. 10. Communication overhead with respect to the number of signatures.

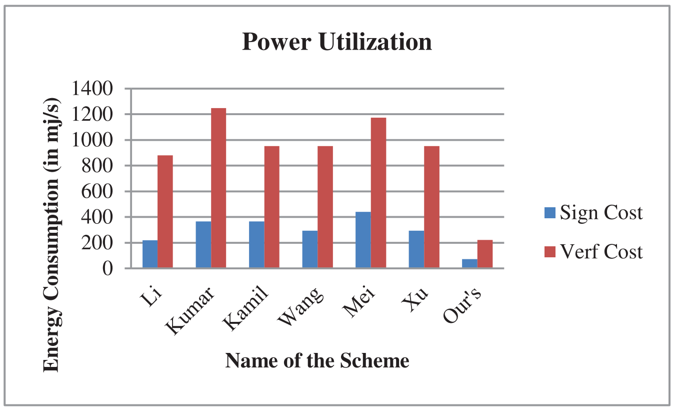

Fig. 11. Power utilization for signature generation and verification of vehicles.

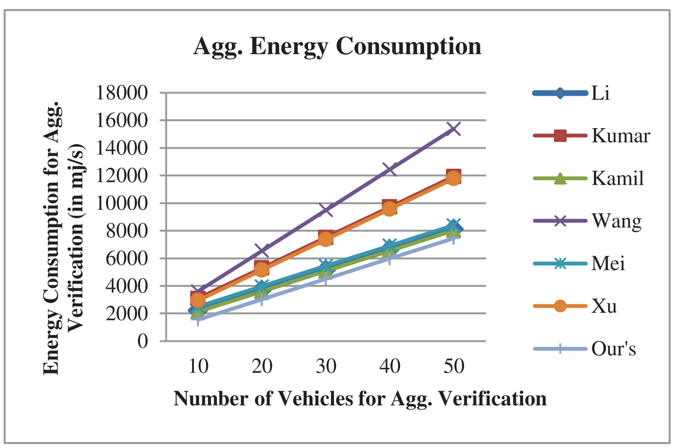

Fig. 12. Power utilization of aggregate signature verification.

our CLAS scheme does not use any pairings, the communication cost of our scheme requires 60 bytes and is significantly less than other existing pairing-based schemes.

From the above discussion, it is clear that the proposed scheme is more efficient when compared with the existing secure CLAS schemes in terms of computational cost, communication complexity, transmission overhead, and power utilization. Also, the proposed scheme attains all types of security requirements for VANETS when comparing to existing schemes. Hence, our proposed scheme is a secure and efficient pairing-free CLAS scheme for VANETS.

VI. Conclusion

In this article, we have presented an efficient and secure CLAS-based authentication scheme for VANETs. The proposed scheme is free from complex certificate management and the key escrow problem. As the proposed aggregate signature scheme allows different individual signatures on different messages from different vehicles and then aggregate into a single signature, this technique simplifies the verification time, computation cost, and bandwidth requirement and storage space at RSU. The proposed CLAS scheme is constructed in a pairing-free environment, which greatly reduces the computation burden than the existing bilinear pairing-based authentication schemes for VANETs. In the random oracle model, the proposed scheme is proven secure and unforgeable under the assumption of the ECDLP is hard. Thus, the proposed scheme can prevent malicious vehicles from disrupting the security features of VANETs. The extensive performance evaluation shows that the proposed CLAS authentication scheme is more efficient in terms of security, computational, and communication point of view. Hence, our pairing-free CLAS-based authentication scheme is more feasible for the VANET environment.

References

[1] S. M. Hatim, S. J. Elias, N. Awang, and Md. Y. Darus, “VANETS and Internet of Things (IoT): A discussion,” *Indonesian J. Elect. Eng. Comput. Sci.*, vol. 12, no. 1, pp. 218–224, 2018.

[2] M. Azees, P. Vijayakumar, and L. J. Deborah, “Comprehensive survey on security services in vehicular ad-hoc networks,” *IET Int. Transp. Syst.*, vol. 10, no. 6, pp. 379–388, Aug. 2016.

[3] H. Lu and L. Jie, “Privacy-preserving authentication schemes for vehicular ad hoc networks: A survey,” *Wireless Commun. Mobile Comput.*, vol. 16, no. 6, pp. 643–655, 2016.

[4] 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.*, vols. 451–452, pp. 1–15, Jul. 2018.

[5] O. Ogundoyin, “An autonomous lightweight conditional privacy-preserving authentication scheme with provable security for vehicular ad-hoc networks,” *Int. J. Comput. Appl.*. vol. 42, no. 12, pp. 196–211, 2020, doi: 10.1080/1206212X.2018.1477320.

[6] P. Kumar, S. Kumari, V. Sharma, X. Li, S. A. Kumar, and S. K. H. Islam, “Secure CLA and CL-AS schemes designed for VANETs,” *J. Supercomput.*, vol. 75, pp. 3076–3098, Jun. 2019, doi: 10.1007/s11227-018-2312-y.

[7] F. Qu, Z. Wu, F. Wang, and W. Cho, “A security and privacy review of VANETs,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 2985–2996, Dec. 2015.

[8] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, *Aggregate and Verifiably Encrypted Signatures From Bilinear Maps* (Lecture Notes in Computer Science), vol. 2656. Heidelberg, Germany: Springer, 2003, pp. 416–432.

[9] J. Chen, H. Yue, and Z. Huang, “Secure certificate-based aggregate signature scheme,” *Comput. Eng. Appl.*, vol. 49, no. 21, pp. 60–64, 2018.

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

[11] L. Cheng, Q. Wen, Z. Jin, H. Zhang, and L. Zhou, “Cryptanalysis and improvement of a certificateless aggregate signature scheme,” *Inf. Sci.*, vol. 295, pp. 337–346, Feb. 2015.

[12] K.-A. Shim, “Security models for certificateless signature schemes revisited,” *Inf. Sci.*, vol. 296, pp. 315–321, Mar. 2015.

[13] 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, Jan. 2013.
[14] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, no. 177, pp. 203–209, 1987.
[15] V. Miller, "Use of elliptic curves in cryptography," in *Proc. Adv. Cryptol. (Crypto)*, 1985, pp. 417–426.
[16] D. De, S. Zeadally, B. Xu, and X. Huang, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 12, pp. 2681–2691, Dec. 2015.
[17] M. Raya and J. Hubaux, "Securing vehicular ad hoc networks," *J. Comput. Security*, vol. 15, no. 1, pp. 39–68, 2007.
[18] Y. Sun, R. Lu, X. Lin, X. Shen, and J. Su, "An efficient pseudonymous authentication scheme with strong privacy preservation for vehicular communications," *IEEE Trans. Veh. Technol.*, vol. 59, no. 7, pp. 3589–3603, Sep. 2010.
[19] J. Li, H. Lu, and M. Guizani, "ACPN: A novel authentication framework with conditional privacy-preservation and non-repudiation for VANETs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 938–948, Apr. 2015.
[20] N.-W. Lo and J.-L. Tsai, "An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks without pairings," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 5, pp. 1319–1328, May 2016.
[21] Y. Xie, L. Wu, J. Shen, and A. Alelaiwi, "EIAS-CP: New efficient identity-based authentication scheme with conditional privacy-preserving for VANETs," *TeleCommun. Syst.*, vol. 65, no. 2, pp. 229–240, 2017.
[22] 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.
[23] Y. Wang, H. Zhong, Y. Xu, J. Cui, and F. Guo, "Efficient extensible conditional privacy-preserving authentication scheme supporting batch verification for VANETs," *Security Commun. Netw.*, vol. 9, no. 18, pp. 5460–5471, 2016.
[24] S. F. Tzeng, S. J. Horng, T. Li, X. Wang, P. H. Huang, and M. K. Khan, "Enhancing security and privacy for Identity-based batch verification scheme in VANET," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3235–3248, Apr. 2017.
[25] J. Cui, J. Zhang, H. Zhong, and Y. Xu, "SPACF: A secure privacy-preserving authentication scheme for VANET with cuckoo filter," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 10283–10295, Nov. 2017.
[26] I. Ali, T. Lawrence, and F. Li, "An efficient identity-based signature scheme without bilinear pairing for vehicle-to-vehicle communication in VANETs," *J. Syst. Archit.*, vol. 103, Feb. 2020, Art. no. 101692. [Online]. Available: https://doi.org/10.1016/j.sysarc.2019.101692
[27] C. Wang, Z. Dai, D. Zhao, and F. Wang, "A novel identity-based authentication scheme for IoV security," *Int. J. Netw. Security*, vol. 22, no. 4, pp. 627–637, 2020, doi: 10.6633/IJNS.906.
[28] 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, Oct. 2015.
[29] J. Li, H. Yuan, and Y. Zhang, "Cryptanalysis and improvement of certificateless aggregate signature with conditional privacy preserving for vehicular sensor networks," IACR, Lyon, France, Rep. 2016/692, 2016. [Online]. Available: https://eprint.iacr.org/2016/692
[30] A. K. Malhi and S. Batra, "An efficient certificateless aggregate signature scheme for vehicular ad-hoc networks," *Discrete Math. Theor. Comput. Sci.*, vol. 17, no. 1, pp. 317–338, 2015.
[31] P. Kumar and V. Sharma, "On the security of certificateless aggregate signature scheme in vehicular ad hoc networks," in *Proc. Soft Comput. Theories Appl. Adv. Intell. Syst. Comput.*, 2018, pp. 715–722.
[32] X. Yang, C. Chen, T. Ma, Y. Li, and C. Wang, "An improved certificateless aggregate signature scheme for vehicular ad-hoc networks," in *Proc. IEEE 3rd Adv. Inf. Technol. Electron. Autom. Control Conf. (IAEAC)*, Chongqing, China, 2018, pp. 2334–2338.
[33] J. Liu, H. Cao, Q. Li, F. Cai, X. Du, and M. Guizani, "A large-scale concurrent data anonymous batch verification scheme for mobile healthcare crowd sensing," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1321–1330, Apr. 2019, doi: 10.1109/JIOT.2018.2828463.
[34] P. Kumar, S. Kumari, V. Sharma, A. K.Sangaiah, J. Wei, and X. Li, "A certificateless aggregate signature scheme for healthcare wireless sensor network," *Sustain. Comput. Inf. Syst.*, vol. 18, pp. 80–89, Jun. 2018.
[35] Y. Zhan and B. Wang, "Cryptanalysis of a certificateless aggregate signature scheme for healthcare wireless sensor network," *Security Commun. Netw.*, vol. 2019, pp. 1–5, Jun. 2019.
[36] N. B. Gayathri, G. Thumbur, P. V. Reddy, and M. Z. U. Rahman, "Efficient pairing-free certificateless authentication scheme with batch verification for vehicular ad-hoc networks," *IEEE Access*, vol. 6, pp. 31808–31819, 2018.
[37] D. Wang and J. Teng, "Probably secure certificateless aggregate signature algorithm for vehicular ad hoc network," *J. Electronica Inf. Technol.*, vol. 40, no. 1, pp. 11–17, 2018.
[38] X. Hu, W. Tan, C. Yu, C. Ma, and H. Xu, "Security anlysis of certificateless aggregate signature scheme in VANETs," in *Proc. 12th Int. Congr. Image Signal Process. BioMed. Eng. Informat.*, Suzhou, China, 2019, pp. 1–6.
[39] 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, Feb. 2019.
[40] I. A. Kamil and S. O. Ogundoyin, "On the security of privacy-preserving authentication scheme with full aggregation in vehicular ad hoc network," *Security Privacy*, vol. 3, no. 3, p. e104, 2020. [Online]. Available: https://doi.org/10.1002/spy2.104
[41] I. A. Kamil and S. O. Ogundoyin, "An improved certificateless aggregate signature scheme without bilinear pairings for vehicular ad hoc networks," *J. Inf. Security Appl.*, vol. 44, pp. 184–200, Feb. 2019.
[42] Y. Zhao, Y. Hou, L. Wang, S. Kumari M. K. Khan, and H. Xiong, "An efficient certificateless aggregate signature scheme for the Internet of Vehicles," *Trans. Emerg. Telecommun. Technol.*. vol. 31, pp. 1–20, May 2020. [Online]. Available: https://doi.org/10.1002/ett.3708
[43] Q. Mei, H. Xiong, J. Chen, M. Yanng, S. Kumari, and M. K. Khan, "Efficient certificateless aggregate signature with conditional privacy preservation in IoV," *IEEE Syst. J.*, early access, Feb. 25, 2020, doi: 10.1109/JSYST.2020.2966526.
[44] Z. Xu, D. He, N. Kumar, and K.-K. R. Choo, "Efficient certificateless aggregate signature scheme for performing secure routing in VANETs," *Security Commun. Netw.*, vol. 2020, Feb. 2020, Art. no. 5276813.
[45] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *J. Cryptol.*, vol. 13, no. 3, pp. 361–369, 2000.
[46] P. Barreto, H. Y. Kim, and B. Lynn, *Efficient Algorithms for Pairing-Based Cryptosystems* (Lecture Notes in Computer Science), vol. 2442. Heidelberg, Germany: Springer, 2002, pp. 354–368.
[47] X. Cao, W. Kou, and X. Du, "A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges," *Inf. Sci.*, vol. 180, no. 15, pp. 2895–2903, 2010.
[48] S. H. Tan, S. H. Heng, and B. M. Goi, *Java Implementation for Pairing-Based Cryptosystems* (Lecture Notes in Computer Science), vol. 6019. Heidelberg, Germany: Springer, 2010, pp. 188–198.
[49] *MIRACLLibrary*. Accessed: Jun. 13, 2020. [Online]. Available: http://certivox.org/display/ext/miracl