IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 6, 2018, accepted May 30, 2018, date of publication June 18, 2018, date of current version June 29, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2845464

Efficient Pairing-Free Certificateless Authentication Scheme With Batch Verification for Vehicular Ad-Hoc Networks

N. B. GAYATHRI[1], GOWRI THUMBUR[2], (Senior Member, IEEE), P. VASUDEVA REDDY[1], AND MUHAMMAD ZIA UR RAHMAN[3], (Senior Member, IEEE)

[1]Department of Engineering Mathematics, Andhra University, Visakhapatnam 530009, India
[2]Department of Electronics and Communication Engineering, Gandhi Institute of Technology and Management, Visakhapatnam 530045, India
[3]Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Guntur 522502, India

Corresponding author: P. Vasudeva Reddy (vasucrypto@andhrauniversity.edu.in)

This work was supported by the Department of Science and Technology, Ministry of Science and Technology, India, through the Women Scientist Scheme under Grant SR/WOS-A/PM-1033/2014(G).

ABSTRACT The continuous progress of the wireless communication technology provides an intelligent and efficient transportation system through vehicular ad-hoc networks (VANETS) to mitigate traffic jams and road fatalities, which improves safety of passengers and traffic flow. Many researchers, vehicle manufacturers, and telecommunication industries are working on VANETS to construct the next generation transport system. In VANETS, vehicles, equipped with wireless devices, exchange the traffic-related information with other vehicles and the fixed road side units (RSUs). The information shared between vehicles and RSUs in VANETS must be secure. For secure communications in VANETS, many cryptographic schemes were proposed in different settings, and most of the schemes are using bilinear pairings over elliptic curves. But the computation of a bilinear pairing is very expensive. Also the verification of signatures/messages sent by vehicles increases the computational workload on RSUs. In order to improve computational efficiency and transmission overhead, in this paper, we present an efficient pairing-free certificateless authentication scheme with batch verification for VANETS. We designed the scheme in pairing-free environment which improves the communication and computational efficiency. The proposed scheme supports batch verification, which significantly reduces the computational workload on RSUs in VANETS. The proposed scheme is proven secure in the random oracle model and meets the security requirements, such as authenticity, integrity, traceability, anonymity, and revocation. We compared our scheme with well-known existing schemes, and the efficiency analysis shows that the proposed scheme is more efficient.

INDEX TERMS Authentication, batch verification, digital signature, elliptic curve discrete logarithm problem, intelligent transportation system, vehicular ad hoc networks.

I. INTRODUCTION

The advancements in wireless communication technology lead us to introduce the intelligent transportation system in metropolitan cities to manage the traffic caused by thousands of vehicles. These intelligent transportation systems are built using ''Smart vehicles'', equipped with On Board Units (OBUs) and wireless communication devices. These OBUs have the ability to communicate with other OBUs on the vehicles and with the Road Side Units (RSUs), which are located on the road. With these units, two types of communications are possible: Vehicle to Vehicle (V2V) communications where OBUs communicate each other and Vehicle-to-Infrastructure (V2I) communication where OBUs communicate with RSUs. These communications are depicted in Fig. 1. These communications will be monitored by a Trust Authority (TA). The secure and trustful communications plays a crucial role in Vehicular ad-hoc networks (VANETS). Secure communications in VANETS enhances the traffic management, and mitigates traffic accidents, traffic jam, parking difficulty by providing safety related information (to other

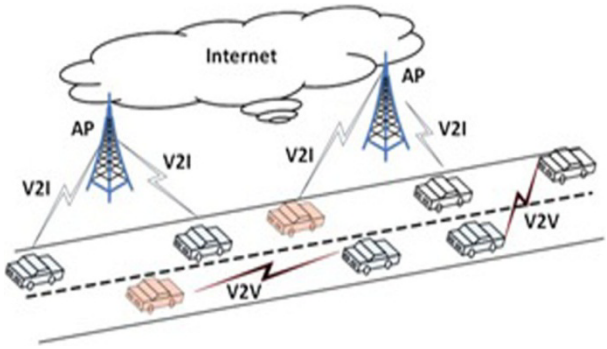

FIGURE 1. VANET Architecture.

vehicles) such as traffic signal violation warning, curve speed warning, pedestrian cross warning, post crash notifications, current position of roads and intersections etc. Hence the information shared in VANETS must be satisfied with several cryptographic security requirements such as authentication, integrity, privacy, non-repudiation, traceability, anonymity, of which authentication and privacy preservation are essential for effective security. If the information shared in VANETS does not meet the cryptographic security standards then adversary may target these communications to various kinds of attacks such as eavesdropping, jamming, interference etc. and destroy the network. Hence, there is a need of cryptographic protection to provide secure communication among vehicles. This attracted the attention of researchers to develop the cryptographic protection of messages among the vehicles [1]–[4]. Digital signature is a cryptographic mechanism which provides the authentication and integrity of messages exchanged in VANETS. Digital signature on each message by OBU, before sending it to other vehicles or RSUs, ensures identity authentication, message integrity, entity-authentication, privacy, non-repudiation in VANETS.

A. RELATED WORK

Many sophisticated security schemes have been proposed in the literature to ensure that all the information exchanged in VANET is authenticated. Some Public Key Infrastructure (PKI) authentication schemes [5], [6] for VANETS have been proposed. Though digital signature in conventional PKI provides integrity and authentication, the maintenance of certificates for vehicles public keys incurs huge computation and communication overhead. To overcome the difficulties in conventional PKI, many Identity-based (ID-based) authentication schemes have appeared in the literature [7]–[21]. In 2010, J. Sun *et al.* [7] presented an identity-based security system for user privacy in VANETS. Later, in 2011, C. Zhang *et al.* [8] proposed an identity based batch verification scheme with group testing for VANETS. Later Lee *et al.* [9] presented an improved scheme to overcome flaws of Zhang *et al.* scheme [8] by proving [8] is vulnerable to the replaying attack and does not achieve signature non-repudiation. In 2012, K. A. Shim [10] proposed an ID-based conditional privacy preserving authentication scheme (CPAS) for secure V2I communication in VANETS. In 2013, S. J. Horng *et al.* [11] proposed a batch verification authentication scheme in VANET for secure pseudonyms. In 2014, J. Zhang *et al.* [12] proved that Lee *et al.* [9] scheme is insecure and presented an improved scheme with same efficiency. In 2015, D. He *et al.* [13] proposed an efficient identity based CPAS for VANETS.

In 2016, M. Azees *et al.* [14] presented the state-of-the-art by reviewing VANET system model, characteristics of VANETS and various security services are discussed for VANETS. This paper summarizes all security attacks and presented related possible counter measures. In 2016, N.W. Lo *et al.* [15] developed a new ID-based signature scheme using ECC for CPAS. This scheme requires less communication bandwidth to transmit the signed message. In 2016, Y. Liu *et al.* [16] presented an efficient anonymous authentication protocol based on signature with message recovery to improve the efficiency of the system. In 2016, H. Lu *et al.* [17] presented a survey on privacy preserving authentication schemes for VANETS. In 2016, Y. Wang *et al.* [18] proposed an extensible conditional privacy preserving pseudo identity based authentication scheme which satisfy batch verification. Also in this scheme, the pseudo identities and the corresponding private keys are generated by PKG alone. In 2017, S. F. Tzeng *et al.* [19] proposed an efficient ID-based batch verification scheme for VANETS and pointed some security risks. X. Hu *et al.* [20] proposed a secure ID-based batch verification scheme without pairings for VANETS by improving S. F. Tzeng *et al.* scheme [19]. In 2017, J. Cui *et al.* [21] proposed the SPACF scheme and uses cuckoo filter and binary search method in batch verification phase for efficiency. All these schemes are designed in identity-based frame work. Though this ID-based system eliminates the difficulties in PKI, it suffers from inherent key escrow problem. To overcome the certificate management and key escrow problems, Al-Riyami [22] introduced the Certificateless (CLS) based mechanism in 2003. Advantages of certificateless based setting attracted the researchers to design various cryptographic schemes in this framework. Many CLS signatures have been evolved in literature for various applications [23]–[26]. However, one cannot adopt these signature schemes directly for authentication in VANETS due to various security requirements. To meet the security requirements in VANETS, very few CLS authentication schemes have appeared in literature [27]–[31]. All these schemes are using Aggregation procedure based on pairings. Aggregation is a technique where all the valid signatures can be aggregated by a third party and this aggregated signature can be verified. But sometimes it is required to verify multiple signatures in a single instance rather than aggregating them, for VANETS. Here comes the concept of Batch verification. Batch verification is a process where multiple signatures can be verified at a time instead of verifying them one by one. Now we review the literature on CLS signature schemes for VANETS in detail. In 2014, A. Malip *et al.* [27] presented a novel certificateless privacy preserving authen-

tication announcement protocol for VANETS. In 2015, A. K. Malhi *et al.* [28] proposed a new efficient certificateless aggregate signature scheme for VANETS and proved its security in random oracle model under the assumption of CDHP is intractable. Also the proposed scheme is computationally more efficient due to its constant pairing operations. In 2015, S. J. Horng *et al.* [29] proposed a conditional privacy preserving aggregate signature scheme for V2V communication in VANETS. They also mentioned that their scheme supports batch verification. This scheme is based on CLS setting with pairings and is the only scheme for VANETS in CLS setting that supports batch verification. But J. Li *et al.* [30] presented a cryptanalysis on S. J. Horng *et al.* scheme [29] by discussing the vulnerabilities of malicious-but-passive KGC attack and presented an improved scheme. Recently, in 2018, P. Kumar *et al.* [31] proposed CLS and CL-AS schemes designed for VANETS using bilinear pairings. Thus there is a need to design a certificateless authentication scheme for VANETS that supports batch verification process.

B. MOTIVATION

Moreover, all the above CLS signature schemes are designed using bilinear pairings over elliptic curves. The time consuming cryptographic operation is pairing operation and is more expensive than the evaluation of a scalar multiplication in elliptic curve. For example, ECC with 224 bit keys provides the same level of security as RSA with 2048 bit keys. Thus ECC has become popular since it provides higher security with smaller keys in size. This smaller key size improves the computational and communicational efficiency, storage capacity, bandwidth efficiency.

Also, in V2I communications, RSUs need to verify large number of signatures that are generated by OBUs. But verifying these signatures sequentially requires lot of computational cost and time. In VANETS, for every100-300 ms, hundreds of messages will be send to RSUs. To reduce the computational cost and time in verification process, Batch Verification technique is used to verify the signatures simultaneously instead of verifying each signature individually. In these VANET based applications, the capacity of bandwidth and computational resources are limited. The evaluation of pairing operation by RSU requires large computing resources and time. For e.g. the evaluation of one pairing operation is 20 times with that of scalar multiplication. In this regard, to further improve the computational efficiency in VANET based applications, it is required to improve the efficiency by eliminating the pairing operations. This motivated us to design a pairing free certificateless authentication scheme for VANETS. To the best of our knowledge, this is the first pairing free certificateless authentication scheme designed for VANETS.

C. OUR CONTRIBUTIONS

The main contributions in this paper are as follows.

i) We proposed a CLS based authentication scheme for secure communication in VANETS.
ii) The construction of our CLS authentication scheme does not use any pairing operation over elliptic curves.
iii) Our CLS authentication scheme is secure against Forgeability, Traceability, Anonymity and Revocation.
iv) Compared with existing related schemes in the literature, our scheme improves the computational efficiency.
v) Our scheme uses batch verification technique to verify multiple signatures in a single instance, which significantly mitigates the computational workload on RSUs.

D. STRUCTURE OF THE PAPER

The rest of the paper is arranged as follows. In Section II we presented preliminaries, syntax and security model for our scheme. In Section III we presented our CLS authentication scheme for VANETS and security analysis. Section IV presents efficiency analysis of the proposed scheme. Finally we presented the conclusions of this paper in Section V.

II. SYNTAX AND SECURITY MODEL

This section presents some preliminaries related to elliptic curves and ECDLP. This section also presents system architecture, frame work and security requirements of the proposed scheme.

Notations and their meanings which we used throughout this paper are tabulated in Table 1.

TABLE 1. 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 Parameter.
V_i	i^{th} Vehicle
(T_{Pub}, s_1)	A public and private key pair of TA
(P_{Pub}, s)	A public and private key pair of KGC
(PK_i, SK_i)	Public and private key pair of the vehicle
D_i	Partial private key of the vehicle V_i
RID_i	The real identity of the vehicle V_i
$ID_i = (ID_{1i}, ID_{2i}, T_i)$	Pseudo identity of the vehicle V_i
T_i	Valid time period of pseudo identity
t_i	Current time stamp
$h, H, H_0, H_1, H_2, H_3, H_4$	Cryptographic one way hash functions
$\mathrm{ADV}_1, \mathrm{ADV}_2$	Type-I and Type-II adversaries of the authentication scheme respectively.
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

A. PRELIMINARIES

1) ELLIPTIC CURVE GROUP

An elliptic curve E over a prime finite field F_P, is defined by an equation $y^2 = (x^3 + ax + b)$, where $a, b \in F_P$ and $4a^3 + 27b^2 \neq 0$. Then $G = \{(x, y) : x, y \in F_P,$

$E(x, y) = 0\} \cup \{O\}$ is the additive elliptic curve group where O is the point at infinity [32].

2) ELLIPTIC CURVE DISCRETE LOGARITHM PROBLEM (ECDLP)

Given $P, Q \in G$, to find an integer $x \in Z_q^*$, such that $Q = xP$. Computation of x from P and Q is computationally hard by any polynomial-time bounded algorithm.

B. SYSTEM ARCHITECTURE

Our VANET structure consists of four entities: TA, Key Generation Centre (KGC), RSU and OBU.

1) **TA:** It is completely trusted authority and can never be compromised. It is responsible to register the vehicles and RSUs with itself. TAs and RSUs communicate using a secure Transport Layer Security protocols. Since the vehicles initially registers with TA, TA alone can have the knowledge of the real identities (RID). This real identity can be recovered by TA, from the corresponding pseudo identity. In case of malicious vehicle, the TA will trace the RID from the corresponding pseudo identity and no other party can trace this real identity.
2) **KGC:** It is a trusted third party, generates partial private keys for vehicles. KGC and TA are always trusted and can never collude with each other.
3) **RSU:** RSU act as a bridge between TA, KGC and OBUs. RSU is connected with TA and KGC by wire whereas RSUs are connected with OBUs by wireless channel. Pseudo identity is generated by RSU under whose coverage is the vehicle requesting for the pseudo identity. The pseudo identities are allocated to vehicles each time a new RSU is encountered. To reduce the consumption of network bandwidth due to frequent updation of pseudo identities under each RSU, we assume that RSUs may combine to form the autonomous networks. We assume that autonomous network is comprised of 4 RSUs in scarcely populated area and 2 RSUs in densely populated areas.
4) **OBU:** On board units are embedded in vehicles, and broadcast the traffic related messages, location identity and driving status etc. This device has its own clock for generating correct time stamp and is able to run on its own battery. For this, all TA, KGC, RSU and OBU have roughly synchronized clocks. OBUs communicate with each other and also with RSUs too. In the following, Fig. 2 explains the steps involved in the proposed authentication scheme for VANETS.

(1) Vehicle registration with TA.
(2) TA generates a Token and preloads in vehicles OBU.
(3) Vehicle requests for Partial private key.
(4) KGC generates Partial private key.
(5) Vehicle generates Public/Secret key pair.
(6) RSU generates pseudo identity after a request from vehicle.
(7) Vehicle to Infrastructure (V2I) communication.
(8) Vehicle to Vehicle (V2V) communication.

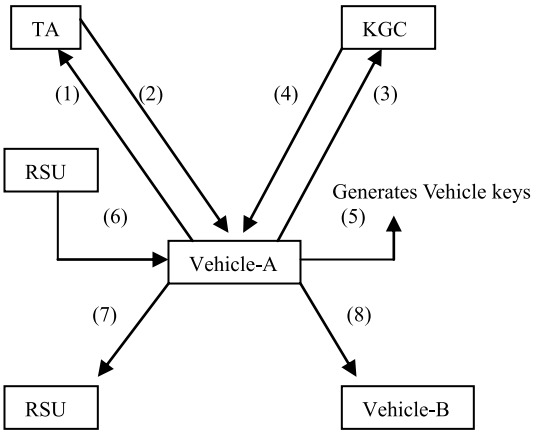

FIGURE 2. Steps involved in proposed authentication system for VANETS.

C. SCHEME FRAMEWORK

Our scheme consists of the following seven algorithms.

1) SystemInitialization: This algorithm is performed by TA and KGC, by taking the security parameter $n \in Z^+$ as input. This algorithm generates the master public/secret key pair and publishes the list of public parameters as *params*. Also TA takes some *params*, real identity as input and calculates vehicles ticket *Token* and generates a signature on *Token*.
2) PartialKeyGen: It is performed by KGC that takes the vehicles *Token* as input and calculate its partial private key.
3) VehicleKeyGen: Vehicle takes *Token* generated by TA and partial private key generated by KGC as input to generate vehicles public/secret key pair.
4) PseudoIdentityGen: This algorithm is performed by RSU by taking a vehicles ticket *Token*as input and outputs its pseudo identity.
5) SignatureGeneration: It is performed by each vehicle, takes a message $m \in \{0, 1\}^*$, public/secret key pair, partial private key of the vehicle and its pseudo identity as input and outputs a signature σ.
6) SignatureVerification: The individual verification is performed by each vehicle that takes system parameters *params*, pseudo identity, message with current time stamp, signature σ as input and outputs true if the signature is valid and false otherwise.
7) BatchVerification: Batch verification is performed by RSU and process is similar to individual verification.

D. SECURITY REQUIREMENTS

The following are the basic security requirements for secure communications in VANETS.

1) MessageAuthentication: The message authenticity ensures that the received message is indeed transmitted by a vehicle which is claimed to done so.
2) Integrity: It ensures that the message has not been modified or forged or dropped while it is communicated from the sender to the receiver.

3) Non − repudiation: Authenticated vehicles could not deny messages after sending them to other vehicles in VANETS.
4) Traceability: TA alone can identify the real identity of the sender by taking its pseudo identity and can identify the malicious messages that are sent by vehicles.
5) Anonymity: Other vehicles and adversaries in VANET cannot identify the senders' real identity either by analyzing multiple messages sent by the same vehicle or by its pseudo identity.
6) Revocation: TA can terminate the communication when the vehicle is confirmed to be a malicious vehicle. Also TA updates the Revocation list by including the malicious vehicles and sends this list to KGC and RSU.

III. PROPOSED SCHEME

In this section we propose an efficient CLS based authentication scheme and its security.

A. PROPOSED SCHEME

As discussed in section II, the proposed scheme consists of the following algorithms.

1) SystemInitialization: TA and KGC set up the system parameters for each RSU and OBU as follows.
   - TA generates the system parameters by taking the security parameter $n \in Z^+$ as input. TA also chooses a group G of prime order q, a generator P of G, and chooses a random $s_1 \in Z_q^*$ as its master secret key and set $T_{Pub} = s_1 P$ as its master public key.
   - KGC selects a random $s \in Z_q^*$ as its master secret key and set its master public key as $P_{Pub} = sP$.
   - TA and KGC selects hash functions $h, H, H_0, H_1, H_2, H_3, H_4 : \{0,1\}^* \rightarrow Z_q^*$, and publish the system parameters as *params*.

$$params = \{q, G, P, P_{Pub}, T_{Pub}, H', H, H_0, H_1, H_2, H_3, H_4\}.$$

- The vehicle $V_i's$ OBUs are secretly preloaded with his parameters $\{Token_i, Sig(Token_i; s_1)\}$ by TA, where $Token_i = \{RID_i \oplus H(\beta T_{Pub}), \beta P, Q_i\}$ for some $\beta \in Z_q^*$, and $Q_i = H_0(RID_i)$, RID_i is the vehicles real identity. Note that s_1 is known only to TA and s is known only to KGC.

2) PartialKeyGen: When a vehicle V_i requests for partial private key, KGC does as follows.
   - KGC will check the revocation list, which is sent by TA through a secure channel, to confirm whether vehicle V_i is revoked. If the vehicle has not been revoked, KGC will verify the signature $Sig(Token_i; s_1)$ by the public key T_{Pub} of TA. If the signature is valid, then KGC generates a partial private key.
   - KGC takes Q_i, system parameters, a random number $r_i \in Z_q^*$, computes $R_i = r_i P, h_{1i} = H_1(Q_i, R_i, P_{Pub})$ and $d_i = r_i + sh_{1i} \bmod q$. KGC forwards the partial private key $D_i = (d_i, R_i)$ to the vehicle V_i. The vehicle V_i can validate D_i by verifying $d_i P = R_i + h_{1i} P_{Pub}$.

3) VehicleKeyGen: Vehicle V_i runs this algorithm by taking Q_i and chooses a random $x_i \in Z_q^*$ as his secret value and sets $X_i = x_i P$. Set $PK_i = (X_i, R_i)$ as public key and $SK_i = (d_i, x_i)$ as its secret key.
4) PseudoIdentityGen: When a vehicle V_i enters in a region of RSU (j), vehicle V_i requests for pseudo identity. Vehicle V_i submits $\{Token_i, Sig(Token_i; s_1)\}$ to RSU. RSU (j) will check the revocation list, which is sent by TA through a secure channel, to confirm whether vehicle V_i is revoked. If the vehicle has not been revoked, RSU (j) will verify the signature $Sig(Token_i; s_1)$ by the public key T_{Pub} of TA. If the signature is valid, then RSU generates a pseudo identity as follows.
   - RSU selects a random $k_i \in Z_q^*$ and uses $Token_i = \{RID_i \oplus H(\beta T_{Pub}), \beta P, Q_i\}$ to compute $ID_{1i} = k_i P$, and $ID_{2i} = RID_i \oplus H(\beta T_{Pub}) \oplus H'(T_i k_i \beta P)$.
   - The pseudo identity of a vehicle V_i is $ID_i = (ID_{1i}, ID_{2i}, T_i)$ where T_i denotes the corresponding pseudo identity's validity period. This pseudo identity is returned to vehicle V_i.
   - SignatureGeneration: 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 its pseudo identity ID_i, secret value SK_i, partial private key d_i to produce the signature as follows.
   - The vehicle chooses $y_{1i}, y_{2i} \in Z_q^*$, a current time stamp t_i and computes Y_{1i}, Y_{2i}, w_i as follows.

$$Y_{1i} = y_{1i} P,$$
$$Y_{2i} = [(y_{2i} x_i + h_{2i} d_i) \bmod q] P_{Pub} = (u_i, v_i),$$
$$w_i = [u_i(y_{1i} + h_{3i} x_i) + h_{4i} d_i] \bmod q,$$

where $h_{2i} = H_2(m_i, ID_i, Y_{1i})$.

$$h_{3i} = H_3(m_i, ID_i, Y_{1i}, R_i, t_i),$$
$$h_{4i} = H_4(m_i, ID_i, Y_{1i}, R_i, t_i).$$

The signature on message m_i is $\sigma_i = (R_i, Y_{1i}, u_i, w_i)$.

5) SignatureVerification: Given a signature σ_i on a message $m_i \| t_i$, corresponding vehicles pseudo identity ID_i and its public key $PK_i = (X_i, R_i)$, then any verifier can verify the signature as follows. Compute $h_{3i} = H_3(m_i, ID_i, Y_{1i}, R_i, t_i)$, $h_{4i} = H_4(m_i, ID_i, Y_{1i}, R_i, t_i)$ and verify whether the following equation holds. $w_i P - u_i(Y_{1i} + h_{3i} X_i) = h_{4i}(R_i + h_{1i} P_{Pub})$. If it holds, accepts the signature$\sigma_i = (R_i, Y_{1i}, u_i, w_i)$. Otherwise rejects.
6) BatchVerification : RSU runs this algorithm by receiving n distinct signatures $(\sigma_i)_{i=1 \text{ to } n}$ on different messages $(m_i \| t_i)_{i=1 \text{ to } n}$, from different vehicles $(V_i)_{i=1 \text{ to } n}$ with corresponding pseudo identities $(ID_i)_{i=1 \text{ to } n}$ and verify the signatures in a single instance as follows.
   - Compute $h_{3i} = H_3(m_i, ID_i, Y_{1i}, R_i, t_i)$,

$$h_{4i} = H_4(m_i, ID_i, Y_{1i}, R_i, t_i), \quad \text{for } i = 1 \text{ to } n.$$

- RSU Chooses $(\delta_i)_{i=1 \text{ to } n} \in Z_q^*$ randomly and verify the equation

$$\sum_{i=1}^{n} (\delta_i w_i P - \delta_i u_i (Y_{1i} + h_{3i} X_i)) = \sum_{i=1}^{n} \delta_i h_{4i} (R_i + h_{1i} P_{Pub}).$$

If the equation holds, RSU accepts the signatures $\sigma_i = (R_i, Y_{1i}, u_i, w_i)fori = 1$ to n.; rejects otherwise.

Note: KGC will never collude with TA and RSU. Hence KGC does not have access to the pseudo identity, and it cannot forge the signature by replacing private key.

B. PROOF OF CORRECTNESS OF THE PROPOSED SCHEME

The correctness of the scheme can be justified as follows.

$$\begin{aligned} & w_i P - u_i (Y_{1i} + h_{3i} X_i) \\ &= [u_i (y_{1i} + h_{3i} x_i) + h_{4i} d_i] P - u_i (Y_{1i} + h_{3i} X_i) \\ &= u_i (Y_{1i} + h_{3i} X_i) + h_{4i} d_i P - u_i (Y_{1i} + h_{3i} X_i) \\ &= h_{4i} d_i P \\ &= h_{4i} (R_i + h_{1i} P_{Pub}). \end{aligned}$$

C. PROOF OF CORRECTNESS OF BATCH VERIFICATION

The correctness of the batch verification can be justified as follows.

$$\begin{aligned} & \sum_{i=1}^{n} (\delta_i w_i P - \delta_i u_i (Y_{1i} + h_{3i} X_i)) \\ &= \sum_{i=1}^{n} (\delta_i [u_i (y_{1i} + h_{3i} x_i) + h_{4i} d_i] P - \delta_i u_i (Y_{1i} + h_{3i} X_i)) \\ &= \sum_{i=1}^{n} (\delta_i [u_i (Y_{1i} + h_{3i} X_i) + h_{4i} d_i P] - \delta_i u_i (Y_{1i} + h_{3i} X_i)) \\ &= \sum_{i=1}^{n} \delta_i h_{4i} d_i P \\ &= \sum_{i=1}^{n} \delta_i h_{4i} (R_i + h_{1i} P_{Pub}). \end{aligned}$$

D. SECURITY ANALYSIS

In the following, we present the security requirements such as authentication, integrity, and non-repudiation of our proposed scheme. These security properties can be achieved through our CLS signature scheme. In the following Theorem 1, we prove the proposed CLS signature scheme is secure against Type I and Type II adversaries.

Theorem 1: In the ROM, the proposed CLS scheme is secure under the ECDLP assumption.

The proof of Theorem 1 follows from lemma 1 and lemma 2.

Lemma 1: In the random oracle model, if there exists a forger ADV_1 *who can forge a signature after making* q_{H_i} *queries to random oracles* H_i *for i-0, 1, 2, 3, 4,* q_{Rpsk} *queries to the* ***Reveal Partial Secret Key*** *extraction oracle,* q_{Cuser} *queries to the* ***Create User*** *request oracle,* q_{Rsk} ***Reveal Secret Key*** *extraction queries and* q_{Sign} ***Sign*** *queries then the ECDLP can be solved.*

Proof: Suppose ADV_1 is a Type I forger against our CLS scheme. We will show how to produce another algorithm ξ which can solve the ECDLP with the help of ADV_1. Suppose ξ receives a challenge $(P, Q = sP)$. Its goal is to compute s. ξ acts as a challenger and answers the queries posed by ADV_1 as follows. Without loss of generality, ξ takes ID^* as target identity of ADV_1 on a message m_i^*.

- ***Initialization Phase:*** Algorithm ξ sets $P_{Pub} = Q = sP$ and gives the system parameters *params* nd master public key to ADV_1 and keeps s secretly.
- ***Queries Phase:*** ADV_1 performs the following oracles in an adaptive manner and the algorithm ξ will answer to these oracles. To avoid the conflict of simulation, ξ need to maintain the initially empty lists $\mathcal{L}_0, \mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3, \mathcal{L}_4, \mathcal{L}_{PSK}, \mathcal{L}_{Cuser}$. These lists are used to keep track of answers to the following queries.

Queries on oracle H_0 $(H_0(ID_i))$: When ADV_1 makes a H_0 query on (ID_i), ξ will search the list $\mathcal{L}_0$ for the tuple (ID_i, Q_i). If such tuple exists in $\mathcal{L}_0$, then ξ returns Q_i. Otherwise, ξ picks a random Q_i and adds to $\mathcal{L}_0$. Finally, ξ returns Q_i.

Queries on oracle H_1 $(H_1(Q_i, R_i, P_{Pub}))$: When ADV_1 makes a H_1 query on (Q_i, R_i, P_{Pub}), ξ will search the list $\mathcal{L}_1$ for the tuple $(Q_i, R_i, P_{Pub}, l_{1i})$. If such tuple exists in $\mathcal{L}_1$, then ξ returns l_{1i}. Otherwise, ξ picks a random l_{1i} and adds to $\mathcal{L}_1$. Finally, ξ returns l_{1i}.

Queries on oracle H_2 $(H_2(m_i, ID_i, Y_{1i}))$: When ADV_1 makes a H_2 query on (m_i, ID_i, Y_{1i}), ξ will search the list $\mathcal{L}_2$ for the tuple $(m_i, ID_i, Y_{1i}, l_{2i})$. If such tuple exists in $\mathcal{L}_2$, ξ returns l_{2i}. Otherwise, ξ picks a random $l_{2i} \in Z_q^*$ and returns l_{2i}. ξ adds $(m_i, ID_i, Y_{1i}, l_{2i})$ to $\mathcal{L}_2$.

Queries on oracle H_3 $(H_3(m_i, ID_i, Y_{1i}, R_i, t_i))$***:*** When ADV_1 makes a H_3 query on $(m_i, ID_i, Y_{1i}, R_i, t_i)$, ξ will search the list $\mathcal{L}_3$ for the tuple $(m_i, ID_i, Y_{1i}, R_i, t_i, l_{3i})$. If such tuple exists in $\mathcal{L}_3$, then ξ gives l_{3i}. Otherwise, ξ chooses a random $l_{3i} \in Z_q^*$ and returns l_{3i}. Finally, ξ adds the tuple $(m_i, ID_i, Y_{1i}, R_i, t_i, l_{3i})$ to the list $\mathcal{L}_3$.

Queries on oracle $H_4(H_4(m_i, ID_i, Y_{1i}, R_i, t_i))$***:*** When ADV_1 makes a H_4 query on $(m_i, ID_i, Y_{1i}, R_i, t_i)$, ξ will search the list $\mathcal{L}_4$ for the tuple $(m_i, ID_i, Y_{1i}, R_i, t_i, l_{4i})$. If such tuple exists in $\mathcal{L}_4$, then ξ gives l_{4i} to ADV_1. Otherwise, ξ picks a random $l_{4i} \in Z_q^*$ and returns l_{4i}. Finally, ξ adds $(m_i, ID_i, Y_{1i}, R_i, t_i, l_{4i})$ to $\mathcal{L}_4$.

Reveal Partial Secret Key Oracle $(PSK(ID_i))$***:*** When ADV_1 makes a query on $PSK(ID_i)$, ξ will search the list $\mathcal{L}_{PSK}$ for the tuples (ID_i, d_i, R_i).

If such tuple exists in $\mathcal{L}_{PSK}$, then ξ returns d_i. Otherwise, if $ID_i \neq ID^*$, ξ chooses $a_i \in Z_q^*$ and sets $d_i = a_i$ and add (ID_i, d_i, R_i) to $\mathcal{L}_{PSK}$ and returns d_i. If $ID_i = ID^*$, ξ aborts.

Create User Oracle $(Cuser(ID_i))$***:*** When ADV_1 makes a query on $Cuser(ID_i)$, ξ will search the list $\mathcal{L}_{Cuser}$ for the tuple (ID_i, x_i, X_i). If such tuple exists in $\mathcal{L}_{Cuser}$, then ξ outputs X_i. Otherwise, ξ do the following.

i) If $ID_i \neq ID^*$, ξ chooses $a_i, b_i, x_i \in Z_q^*$ randomly and sets $R_i = a_iP - b_iP_{Pub}$, $H_1(Q_i, R_i, P_{Pub}) = b_i$ and $X_i = x_iP$.ξ adds (Q_i, R_i, P_{Pub}, b_i) to $\mathcal{L}_1$ and (ID_i, x_i, X_i) to $\mathcal{L}_{Cuser}$.ξ sends X_i to ADV_1.

ii) If $ID_i = ID^*$, ξ chooses $a_i, b_i, x_i \in Z_q^*$ randomly and sets $R_i = a_iP$, $H_1(Q_i, R_i, P_{Pub}) = b_i$ and $X_i = x_iP$.ξ adds (Q_i, R_i, P_{Pub}, b_i) to $\mathcal{L}_1$ and (ID_i, x_i, X_i) to $\mathcal{L}_{Cuser}$.ξ sends X_i to ADV_1.

Note that (R_i, X_i, h_{1i}) generated in this way satisfies the equation $d_iP = R_i + h_{1i}P_{Pub}$.$\xi$ gets Q_i from $\mathcal{L}_0$ list or it quires H_0 oracle to get Q_i.

Reveal Secret Key Oracle $(RSK(ID_i))$***:*** When ADV_1 performs this query on $RSK(ID_i)$, if $ID_i = ID^*$, ξ aborts. Otherwise, if$ID_i \neq ID^*$, ξ finds the tuple (ID_i, x_i, X_i) from the list $\mathcal{L}_{Cuser}$, and sends x_i to ADV_1. If there is no such tuple in $\mathcal{L}_{Cuser}$, ξ asks a query on $Cuser(ID_i)$ to produce (x_i, X_i) and ξ saves these values in $\mathcal{L}_{Cuser}$. Finally ξ returns x_i.

Replace Public Key Oracle $(RPK(ID_i))$***:*** When ADV_1 performs this query on $RPK(ID_i)$, ξ finds (ID_i, x_i, X_i) in $\mathcal{L}_{Cuser}$.ξ replaces $X_i = X_i'$and$x_i = \perp$.

Signing Oracle: When ADV_1 performs this query on (ID_i, m_i, t_i), ξ first makes queries on H_0, H_1, H_2, H_3, H_4 oracles and recovers Q_i and the tuples $(Q_i, R_i, P_{Pub}, l_{1i})$, $(m_i, ID_i, Y_{1i}, l_{2i})$, $(m_i, ID_i, Y_{1i}, R_i, t_i, l_{3i})$, $(m_i, ID_i, Y_{1i}, R_i, t_i, l_{4i})$ from $\mathcal{L}_0, \mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3, \mathcal{L}_4$ respectively and (ID_i, d_i, R_i) from $\mathcal{L}_{PSK}$ and (ID_i, x_i, X_i) from $\mathcal{L}_{Cuser}$.ξ selects a random $r_{1i} \in Z_q^*$ and sets $Y_{2i} = (r_{1i}P)_x = u_i$, $Y_{1i} = -(R_i + l_{1i}P_{Pub})u_i^{-1}l_{4i}$, $w_i = u_ix_i$. Finally, ξ returns $\sigma_i = (R_i, Y_{1i}, u_i, w_i)$ to ADV_1.

Note that $\sigma_i = (R_i, Y_{1i}, u_i, w_i)$ is a valid signature with

$$w_iP - u_i(Y_{1i} + h_{3i}X_i) = h_{4i}(R_i + h_{1i}P_{Pub}). \tag{1}$$

- ***Forgery:*** Finally, ADV_1 out puts a valid signature tuple $\left(ID_i^*, m^*, t^*, \sigma_i^*\right)$ where$\sigma_i^* = (R_i^*, Y_{1i}^*, u_i^*, w_i^*)$.

If $ID_i \neq ID^*$, ξ stops simulation. Otherwise, ξ looks up at $\mathcal{L}_{PSK}$&$\mathcal{L}_{Cuser}$separately. Let $\sigma_i^{(1)} = (R_i, Y_{1i}, u_i^{(1)}, w_i^{(1)})$ denote $\sigma_i = (R_i, Y_{1i}, u_i, w_i)$. From Forking Lemma [33], if we replay of ξ with same random tape but different choice of H_3, H_4, ADV_1 will generate another three $\sigma_i^{(j)} = (R_i, Y_{1i}, u_i^{(j)}, w_i^{(j)})$ for $j = 2, 3, 4$, such that $w_i^{(j)}P - u_i^{(j)}(Y_{1i} + l_{3i}^{(j)}X_i) = l_{4i}^{(j)}(R_i + l_{1i}^{(j)}P_{Pub})$ for $j = 1, 2, 3, 4$. By r_i, x_i, s, r_{1i}, we now denote discrete logarithms of $R_i, X_i, P_{Pub}, Y_{1i}$ respectively, that is

$$R_i = r_iP, \quad X_i = x_iP, \quad P_{Pub} = sP, \quad Y_{1i} = r_{1i}P.$$

Thus we have four linearly independent equations as follows.

$$w_i^{(j)} - u_i^{(j)}(r_{1i} + l_{3i}^{(j)}x_i) = l_{4i}^{(j)}(r_i + l_{1i}^{(j)}s) \quad \text{for } j = 1, 2, 3, 4.$$

Here r_i, x_i, s, r_{1i} are unknown to ξ and can solve these values from the above equations and outputs s as the solution of ECDLP.

Lemma 2: In the random oracle model, if there exists a forger ADV_2 *who can forge a signature after making* q_{H_i} *queries to random oracles* H_i *for i-0,1,2,3,4,* q_{Cuser} *queries to the* ***Create User*** *request oracle,* q_{Rsk} ***Reveal Secret Key*** *extraction queries and* q_{Sign} ***Sign*** *queries, then the ECDLP can be solved.*

Proof: Suppose ADV_2 is a Type II forger against our CLS scheme. We will show how to produce another algorithm ξ which can solve the ECDLP with the interaction of ADV_2. Suppose ξ receives a challenge $(P, Q = \alpha P)$. Its goal is to compute α.ξ acts as a challenger and answers the queries posed by ADV_2 as follows.

- ***Initialization Phase:*** Challenger ξ sets $P_{Pub} = sP$ and produces system parameters *params*.ξ then sends *params* and master secret key to ADV_2.
- ***Queries Phase:*** ADV_2 performs the following oracles in an adaptive manner and the algorithm ξ will answer to these oracles. To avoid the conflict of simulation, ξ need to maintain initially empty lists $\mathcal{L}_0, \mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3, \mathcal{L}_4, \mathcal{L}_{Cuser}$. These lists are used to keep track of answers to the following queries.

Queries on oracle H_0 $(H_0(ID_i))$: When ADV_2 makes a H_0 query on (ID_i), ξ will search the list $\mathcal{L}_0$ for the tuple (ID_i, Q_i). If such tuple exists in $\mathcal{L}_0$, then ξ returns Q_i. Otherwise, ξ picks a random Q_i and adds to $\mathcal{L}_0$. Finally, ξ returns Q_i.

Queries on oracle H_1 $(H_1(Q_i, R_i, P_{Pub}))$: When ADV_2 makes a H_1 query on (Q_i, R_i, P_{Pub}), ξ will search the list $\mathcal{L}_1$ for the tuple $(Q_i, R_i, P_{Pub}, l_{1i})$. If such tuple exists in $\mathcal{L}_1$, then ξ returns l_{1i}. Otherwise, ξ chooses l_{1i} at random and inserts to the list $\mathcal{L}_1$. Finally, ξ sends l_{1i} to ADV_2.

Queries on oracle H_2 $(H_2(m_i, ID_i, Y_{1i}))$: When ADV_2 makes a H_2 query on (m_i, ID_i, Y_{1i}), ξ will search the list $\mathcal{L}_2$ for the tuple $(m_i, ID_i, Y_{1i}, l_{2i})$. If such tuple exists in $\mathcal{L}_2$, ξ returns l_{2i}. Otherwise, ξ picks a random $l_{2i} \in Z_q^*$ and returns l_{2i}.ξ adds $(m_i, ID_i, Y_{1i}, l_{2i})$ to $\mathcal{L}_2$.

Queries on oracle $H_3(H_3(m_i, ID_i, Y_{1i}, R_i, t_i))$***:*** When ADV_2 makes a H_3 query on $(m_i, ID_i, Y_{1i}, R_i, t_i)$, ξ will search the list $\mathcal{L}_3$ for the tuple $(m_i, ID_i, Y_{1i}, R_i, t_i, l_{3i})$. If such tuple exists in $\mathcal{L}_3$, then ξ gives l_{3i}. Otherwise, ξ chooses a

random $l_{3i} \in Z_q^*$ and returns l_{3i}. Finally, ξ adds the tuple $(m_i, ID_i, Y_{1i}, R_i, t_i, l_{3i})$ to the list $\mathcal{L}_3$.

Queries on oracle H_4 $(H_4(m_i, ID_i, Y_{1i}, R_i, t_i))$**:** When ADV_2 makes a H_4 query on $(m_i, ID_i, Y_{1i}, R_i, t_i)$, ξ will search the list $\mathcal{L}_4$ for the tuple $(m_i, ID_i, Y_{1i}, R_i, t_i, l_{4i})$. If such tuple exists in $\mathcal{L}_4$, then ξ gives l_{4i} to ADV_2. Otherwise, ξ picks a random $l_{4i} \in Z_q^*$ and returns l_{4i}. Finally, ξ adds $(m_i, ID_i, Y_{1i}, R_i, t_i, l_{4i})$ to $\mathcal{L}_4$.

- ***Create User Oracle*** $(Cuser(ID_i))$**:** When ADV_2 makes a query on $Cuser(ID_i)$, ξ will search the list $\mathcal{L}_{Cuser}$ for the tuple (ID_i, x_i, X_i). If such tuple exists in $\mathcal{L}_{Cuser}$, then ξ outputs X_i. Otherwise, ξ do the following.

  i) If $ID_i \neq ID^*$, ξ chooses $r_i, x_i \in Z_q^*$ at random and computes $R_i = r_iP, H_1(Q_i, R_i, P_{Pub}) = h_{1i}$ and $X_i = x_iP$.ξ adds $(Q_i, R_i, P_{Pub}, h_{1i})$ to $\mathcal{L}_1$ and (ID_i, x_i, X_i) to $\mathcal{L}_{Cuser}$.ξ sends X_i to ADV_2 as a response.

  ii) If $ID_i = ID^*$, ξ chooses $r_i \in Z_q^*$ at random and sets $R_i = r_iP, H_1(Q_i, R_i, P_{Pub}) = h_{1i}$ and $X_i = Q = \alpha P$.ξ adds $(Q_i, R_i, P_{Pub}, h_{1i})$ to $\mathcal{L}_1$ and $(ID_i, \perp, X_i)$ to $\mathcal{L}_{Cuser}$.ξ outputs X_i to ADV_2 as an answer to this query.

Note that ξ gets Q_i from $\mathcal{L}_0$ list if it exists, else it quires H_0 oracle to get Q_i.

Reveal Secret Key Oracle $(RSK(ID_i))$**:** When ADV_2 performs this query on $RSK(ID_i)$, if $(ID_i = ID^*)$, ξ aborts. Otherwise (if$ID_i \neq ID^*$), ξ finds the tuple (ID_i, x_i, X_i) from the list $\mathcal{L}_{Cuser}$, and returns x_i to ADV_2. If there is no tuple in $\mathcal{L}_{Cuser}$, ξ asks a query on $Cuser(ID_i)$ to produce (x_i, X_i).ξ saves these values in $\mathcal{L}_{Cuser}$, and returns x_i.

Signing Oracle: On receiving a signature query on (ID_i, m_i, t_i), ξ recovers the tuples $(Q_i, R_i, P_{Pub}, l_{1i})$, $(m_i, ID_i, Y_{1i}, l_{2i})$, $(m_i, ID_i, Y_{1i}, R_i, t_i, l_{3i})$, $(m_i, ID_i, Y_{1i}, R_i, t_i, l_{4i})$ from $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3, \mathcal{L}_4$ respectively and (ID_i, x_i, X_i) from $\mathcal{L}_{Cuser}$.

If $ID_i \neq ID^*$, ξ chooses $r_{1i} \in Z_q^*$ at random and sets $Y_{1i} = r_{1i}P, Y_{2i} = ((x_ir_{1i} + l_{2i}d_i) \bmod q)\, P_{Pub} = (u_i, v_i), w_i = l_{4i}d_i + u_i(r_{1i} + l_{3i}x_i)$.

If $ID_i = ID^*$, ξ chooses $r_{1i} \in Z_q^*$ at random and sets $Y_{1i} = -l_{3i}X_i, Y_{2i} = (r_{1i}P)_x = u_i, w_i = l_{4i}d_i$.$\xi$ returns $\sigma_i = (R_i, Y_{1i}, u_i, w_i)$ to ADV_2.

Note that $\sigma_i = (R_i, Y_{1i}, u_i, w_i)$ is a valid signature and satisfies equation (1)

- ***Forgery:*** Finally ADV_2 outputs $ID_i^*, m^*, t^*, \sigma_i^*$ as its forgery where $\sigma_i^* = (R_i^*, Y_{1i}^*, u_i^*, w_i^*)$.

  If $ID_i \neq ID^*$, ξ stops simulation. Otherwise, ξ looks up at $\mathcal{L}_{Cuser}$ separately. Let $\sigma_i^{(1)} = (R_i, Y_{1i}, u_i^{(1)}, w_i^{(1)})$ denote $\sigma_i = (R_i, Y_{1i}, u_i, w_i)$. From Forking Lemma [33], with different choice of H_2, H_3, H_4, ADV_2 can generate a valid signatures $\sigma_i^{(j)} = (R_i, Y_{1i}, u_i^{(j)}, w_i^{(j)})$ for $j = 2, 3$, such that

$$w_i^{(j)}P - u_i^{(j)}(Y_{1i} + l_{3i}^{(j)}X_i) = l_{4i}^{(j)}(R_i + l_{1i}^{(j)}P_{Pub}) \quad \text{for } j = 1, 2, 3.$$

  By r_i, α, s, r_{1i}, we now denote discrete logarithms of $R_i, X_i, P_{Pub}, Y_{1i}$ respectively, that is $R_i = r_iP, X_i = \alpha P, P_{Pub} = sP, Y_{1i} = r_{1i}P$. Thus we have three independent equations as follows. $w_i^{(j)} - u_i^{(j)}(r_{1i} + l_{3i}^{(j)}\alpha) = l_{4i}^{(j)}(r_i + l_{1i}^{(j)}s)$ for $j = 1, 2, 3$. Here r_i, α, r_{1i}, are unknown to ξ and can solve these values from the above equations and outputs α as the solution of ECDLP.

1) *Traceability:* In our proposed authentication scheme, the vehicle can't be traced based on its identity as pseudo identities are used for communication among vehicles. Since pseudo identity $ID_i = (ID_{1i}, ID_{2i}, T_i)$ is a combination of master secret key (s_1) of TA and some chosen secret value (β) of TA, and hence only the one who knows these values can compute real identity RID_i. Hence it does not leak any information about real identity RID_i. If a signature is in dispute, TA can trace the real identity as follows.

$$RID_i = ID_{2i} \oplus H(\beta T_{Pub}) \oplus H'(T_i\beta ID_{1i}).$$

2) *Anonymity:* In our proposed authentication scheme, each vehicle's real identity RID_i is kept secret and pseudo identity is $ID_i = (ID_{1i}, ID_{2i}, T_i)$ assigned for communication which provides the privacy and anonymity in vehicular networks. Since the vehicle uses pseudo identity, which contains $ID_{1i} = k_iP$ and $ID_{2i} = RID_i \oplus H(\beta T_{Pub}) \oplus H'(T_ik_i\beta P)$, so the private information of the vehicle can't be traced. Also the pseudo identities that are allocated to vehicles are updated every time when the corresponding vehicle enters in to the region of next RSU. Hence anonymity has been achieved in our scheme.
3) *Revocation:* In our proposed scheme, TA maintains and updates a list of malicious vehicles and sends the original revocation list to KGC and RSU through a secure channel. Even if a revoked vehicle requests for a partial private key or a pseudo identity, KGC or RSU will never generate them for illegal vehicles.

IV. EFFICIANCY ANALYSIS

This section presents the performances of our CLS authentication scheme in terms of signing cost, verification cost, total computational cost, transmission overhead and security point of view. We compare our scheme with the existing relevant schemes [27]–[31]. We consider the experimental results [34]–[37] to achieve the comparable security with 1024-bit RSA key, where the bilinear pairing (Tate pairing) is defined over the super singular elliptic curve $E/F_p : y^2 = x^3 + x$ with embedding degree 2 and the 160-bit Solinas prime number $q = 2^{159} + 2^{17} + 1$ with 512-bit prime number p

satisfying $p + 1 = 12qr$. The details of these operations and their conversions are presented in Table 2.

TABLE 2. Notations and descriptions of various cryptographic operations and their conversions.

Notations	Description
T_{MM}	Modular multiplication operation in Z_q^*
T_{SM}	Elliptic curve point multiplication, (Scalar multiplication in G_{Adt}), $T_{SM} = 29T_{MM}$
T_{BP}	Bilinear pairing in G_{Mlt}, $T_{BP} = 87T_{MM}$
T_H	Simple hash function which is negligible
T_{MTPH}	Map to point hash function, $1T_{MTPH} = 1T_{SM} = 29T_{MM}$
T_{PA}	Elliptic curve point addition in G_{Adt}, $T_{PA} = 0.12T_{MM}$

A. COMPUTATION COSTS

We now analyze our authentication scheme and compare it with [27]–[31].

A. Malip *et al.* scheme [27] requires three scalar multiplications, two point additions and two map to point hash functions to produce a digital signature and four bilinear pairings and three map to point hash functions for signature verification. Thus, A. Malip *et al.* scheme [27] needs $3T_{SM} + 2T_{PA} + 2T_{MTPH} = 145.24T_{MM}$ for signature generation and $4T_{BP} + 3T_{MTPH} = 435T_{MM}$ for signature verification. Hence the total computational cost of A. Malip *et al.* scheme [27] is $580.24T_{MM}$.

A. K. Malhi *et al.* scheme [28] requires four scalar multiplications and two point additions to produce a digital signature and three bilinear pairings, three scalar multiplications and one point addition for signature verification. Thus A. K. Malhi *et al.* scheme [28] needs $4T_{SM} + 2T_{PA} = 116.24T_{MM}$ for signature generation and $3T_{BP} + 3T_{SM} + 1T_{PA} = 348.12T_{MM}$ for signature verification. Hence the total computational cost of A. K. Malhi *et al.* scheme [28] is $464.36T_{MM}$.

Similarly, S. J. Horng *et al.* scheme [29] requires two scalar multiplications and one point addition to produce a digital signature and three bilinear pairings, one scalar multiplication and one point addition for signature verification. Thus S. J. Horng *et al.* scheme [29] needs $2T_{SM} + 1T_{PA} = 58.12T_{MM}$ for signature generation and $3T_{BP} + 1T_{SM} + 1T_{PA} = 290.12T_{MM}$ for signature verification. Hence the total computational cost of S. J. Horng *et al.* scheme [29] is $348.24T_{MM}$.

J. Li *et al.* scheme [30] requires two scalar multiplications, one point addition and one map to point hash function to produce a digital signature and three bilinear pairings, one scalar multiplication, one point addition and one map to point hash function for signature verification. Thus J. Li *et al.* scheme [30] needs $2T_{SM} + 1T_{PA} + 1T_{MTPH} = 87.12T_{MM}$ for signature generation and $3T_{BP} + 1T_{SM} + 1T_{PA} + 2T_{MTPH} = 348.12T_{MM}$ for signature verification. Hence the total computational cost of J. Li *et al.* scheme [30] is $435.24T_{MM}$.

P. Kumar *et al.* scheme [31] requires four scalar multiplications, one map to point hash function and two point additions to produce a digital signature and four bilinear pairings, three scalar multiplications and two map to point hash functions for signature verification. Thus P. Kumar *et al.* scheme [31] needs $4T_{SM} + 1T_{MTPH} + 2T_{PA} = 145.24T_{MM}$ for signature generation and $4T_{BP} + 3T_{SM} + 2T_{MTPH} = 493T_{MM}$ for signature verification. Hence the total Computational cost of P. Kumar *et al.* [31] is $638.24T_{MM}$.

Since our scheme is pairing free, it requires only two scalar multiplications for signature generation. For signature verification it requires only five scalar multiplications and three point additions. Thus our scheme needs $2T_{SM} = 58T_{MM}$ for signature generation and $5T_{SM} + 3T_{PA} = 145.36T_{MM}$ for signature verification. The computation costs of schemes [27]–[31] are presented in Table 3.

B. TRANSMISSION OVERHEAD

Now we present the comparison of transmission overhead of the four schemes [28]–[31]. Though all these schemes are established on bilinear pairings, the scheme presented in A. K. Malhi *et al.* [28] and P. Kumar *et al.* [31] are established on ECC. To achieve a security level of 80 bits, in pairing as well as in ECC based schemes, we consider various parameters as shown in Table 4.

We evaluate the transmission overhead by considering signature, pseudo identity, current time stamp, public key and partial private key of the vehicle by excluding the message.

In A.K. Malhi et al scheme [28], the vehicle sends the signature $\sigma_{ijk} = (U_i, V_{ijk}) \in G$, pseudo identity $(PS_j, PS_{1j}) \in G$, public key $P_i \in G$, and partial private key $pp_i \in G$. The total transmission cost is $6|G| + \left|Z_q^*\right| = 2080$bits. Similarly, in S. J. Horng *et al.* scheme [29], and J. Li *et al.* scheme [30], the vehicle sends $\sigma_i = (R_i, S_i) \in G_1$, pseudo identity $ID_i = (ID_{1i}, ID_{2i}, T_i) \in G_1$, public key $vpk_i \in G_1$, and partial private key $psk_i \in G_1$. The total transmission cost is $5|G_1| + \left|Z_q^*\right| + 32 = 5312$bits. In P. Kumar *et al.* scheme [31], the vehicle sends $(U_i, V_{ijk}) \in G$, pseudo identity $(PS_j, PS_{1j}) \in G$, public key $P_i \in G$, and partial private key $pp_i \in G$. The total transmission cost is $6|G| + \left|Z_q^*\right| = 2080$bits. In our proposed scheme, the vehicle sends the signature $\sigma_i = (R_i, Y_{1i}, u_i, w_i) \in G$, pseudo identity $ID_i = (ID_{1i}, ID_{2i}, T_i) \in G$, public key $X_i \in G$, and partial private key $d_i \in Z_q^*$. The total transmission cost is $4|G| + 4\left|Z_q^*\right| + 32 = 1952$bits.

The following Table 5 presents the total transmission overhead of all schemes in terms of sending a single message and n messages.

From Table 3, we observe that the computation cost of our authentication scheme is $203.36T_{MM}$, and is 56.2% less than A. Kaur *et al.* scheme [28], 41.6% less than S. J Horng *et al.* scheme [29], 53.27% less than J. Li *et al.* scheme [30], 64.95% less than A. Malip *et al.* Scheme [24]

TABLE 3. Comparison of the proposed scheme with the related schemes.

Scheme	Signing Cost	Verification Cost	Total Cost	Without Pairing	Secure	Signature Scheme Supports
A. Malip et al. [27]	$3T_{SM}+2T_{PA}+2T_{MTPH}$	$4T_{BP}+3T_{MTPH}$	$580.24T_M$	No	Yes	Aggregation
A. K. Malhi et al. [28]	$4T_{SM}+2T_{PA}$	$3T_{BP}+3T_{SM}+1T_{PA}$	$464.36T_{MM}$	No	Yes	Aggregation
S. J. Horng et al. [29]	$2T_{SM}+1T_{PA}$	$3T_{BP}+1T_{SM}+1T_{PA}$	$348.24T_{MM}$	No	No	Aggregation, Batch Verification
J. Li et al. [30]	$2T_{SM}+1T_{PA}+1T_{MTPH}$	$3T_{BP}+1T_{SM}+1T_{PA}+2T_{MTPH}$	$435.24T_M$	No	Yes	Aggregation
P. Kumar et al. [31]	$4T_{SM}+1T_{MTPH}+2T_{PA}$	$4T_{BP}+3T_{SM}+2T_{MTPH}$	$638.24T_{MM}$	No	Yes	Aggregation
Our Scheme	$2T_{SM}$	$5T_{SM}+3T_{PA}$	$203.36T_M$	Yes	Yes	Batch Verification

TABLE 4. 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	$E: y^2=x^3+x \bmod p$	$\hat{e}: G_1\times G_1\to G_T$	$G_1(P)$	$\lvert p\rvert=512$ bits (64 bytes)	$q=160$ bits	$\lvert G_1\rvert=1024$ bits
ECC	$E: y^2=x^3+ax+b \bmod p$ where $a,b\in Z_q^*$.	Without Pairing	$G(P)$	$\lvert p\rvert=160$ bits (20 bytes)	$q=160$ bits	$\lvert G\rvert=320$ bits

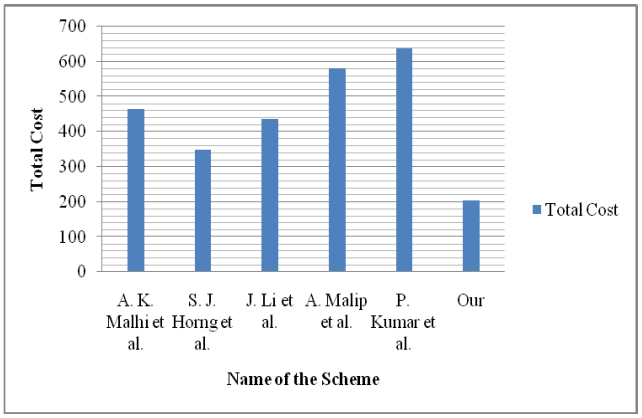

FIGURE 3. Graphical presentation of total computation cost.

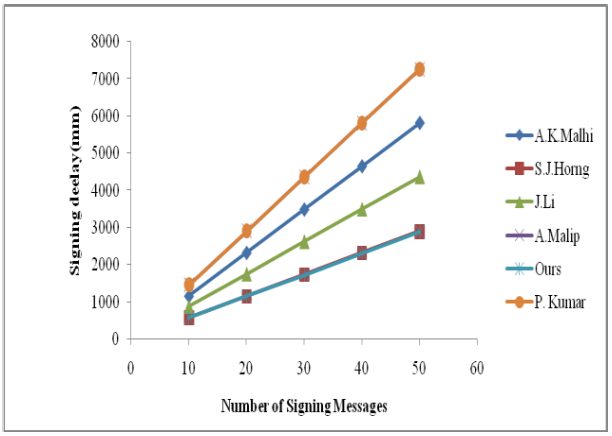

FIGURE 4. Delay in signing messages with respect to the number of messages.

and 68.13% less than P. Kumar *et al.* Scheme [31]. Also comparison of our proposed scheme with the other schemes is presented graphically in Fig. 3. The graph clearly indicates that our scheme is more efficient than the existing schemes. Fig. 4 and Fig. 5 presents the signing and verification delay with respect to number of messages. In Fig. 4, the proposed scheme and S J. Horng *et al.* scheme [29] are having almost same signing delay. But S. J. Horng *et al.* scheme is insecure. Hence, the signing delay of the proposed scheme is smaller than the existing schemes. In Fig. 5, A. Malip *et al.* [27] and J. Li *et al.* [30] schemes verification delay is same. Clearly our scheme yields much less verification delay when comparing to other schemes. From Figs. 4 and 5, we can observe that the slope of the proposed scheme is lower than all other schemes.

TABLE 5. Comparison of the transmission overhead.

Scheme	Sending a message	Sending n messages
A. Kaur et al. [28]	260 bytes	260n bytes
S.J. Horng et al. [29]	664 bytes	664n bytes
J.Li et al. [30]	664 bytes	664n bytes
P.Kumar et al. [31]	260 bytes	260n bytes
Our Scheme	244 bytes	244n bytes

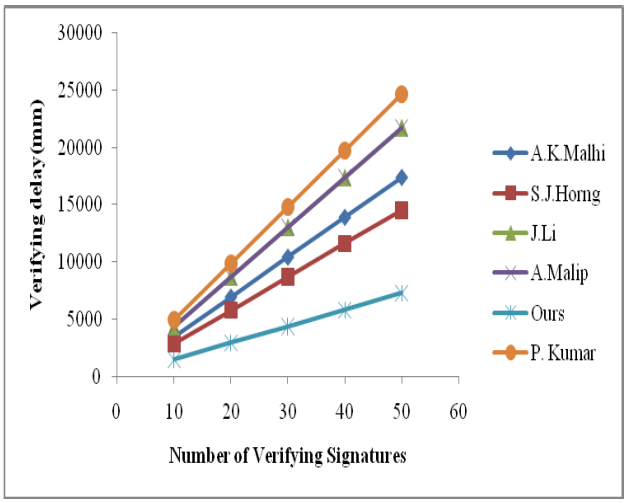

FIGURE 5. Delay in verifying messages with respect to the number of messages.

Hence, of all schemes in the literature, the proposed scheme is more efficient in terms of computational complexity.

V. CONCLUSIONS

In this paper, we have presented an efficient certificateless authentication scheme supporting batch verification for VANETS. The proposed scheme is designed without using bilinear pairings over elliptic curves. The proposed scheme is secure against authentication, integrity, privacy, non-repudiation, traceability, anonymity and revocation. Our scheme uses batch verification technique to verify multiple signatures in a single instance, which significantly mitigates the computational workload on RSUs. The efficiency analysis shows that our authentication scheme is computationally more efficient than the well-known existing schemes. Thus, the proposed scheme can be applied in practice.

REFERENCES

[1] L. Zhang, Q. Wu, A. Solanas, and J. Domingo-Ferrer, "A scalable robust authentication protocol for secure vehicular communications," *IEEE Trans. Veh. Technol.*, vol. 59, no. 4, pp. 1606–1617, May 2010.

[2] T. W. Chim, S. M. Yiu, L. C. K. Hui, and V. O. K. Li, "SPECS: Secure and privacy enhancing communications schemes for VANETs," *Ad Hoc Netw.*, vol. 9, no. 2, pp. 189–203, 2011.

[3] X. Zhu, S. Jiang, L. Wang, and H. Li, "Efficient privacy-preserving authentication for vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 2, pp. 907–919, Feb. 2014.

[4] F. Wang, Y. Xu, H. Zhang, Y. Zhang, and L. Zhu, "2FLIP: A two-factor lightweight privacy-preserving authentication scheme for VANET," *IEEE Trans. Veh. Technol.*, vol. 65, no. 2, pp. 896–911, Feb. 2016.

[5] X. Lin, X. Sun, P.-H. Ho, and X. Shen, "GSIS: A secure and privacy-preserving protocol for vehicular communications," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3442–3456, Nov. 2007.

[6] 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.

[7] J. Sun, C. Zhang, Y. Zhang, and Y. Fang, "An identity-based security system for user privacy in vehicular ad hoc networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 9, pp. 1227–1239, Sep. 2010.

[8] C. Zhang, P.-H. Ho, and J. Tapolcai, "On batch verification with group testing for vehicular communications," *Wireless Netw.*, vol. 17, no. 8, pp. 1851–1865, 2011.

[9] C.-C. Lee and Y.-M. Lai, "Toward a secure batch verification with group testing for VANET," *Wireless Netw.*, vol. 19, no. 6, pp. 1441–1449, 2012.

[10] 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.

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

[12] J. Zhang, M. Xu, and L. Liu, "On the security of a secure batch verification with group testing for VANET," *Int. J. Netw. Secur.*, vol. 16, no. 5, pp. 355–362, 2014.

[13] D. He, S. Zeadally, B. Xu, and X. Huang, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 12, pp. 2681–2691, Dec. 2015.

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

[15] 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.

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

[17] H. Lu and J. Li, "Privacy-preserving authentication schemes for vehicular ad hoc networks: A survey," *Wireless Commun. Mobile Comput.*, vol. 16, no. 6, pp. 643–655, 2016.

[18] Y. Wang, H. Zhong, Y. Xu, J. Cui, and F. Guo, "Efficient extensible conditional privacy-preserving authentication scheme supporting batch verification for VANETs," *Secur. Commun. Netw.*, vol. 9, no. 18, pp. 5460–5471, 2016.

[19] 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 VANETs," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3235–3248, Apr. 2017.

[20] X. Hu, J. Wang, H. Xu, Y. Liu, and X. Zhang, "Secure and pairing-free Identity-based batch verification scheme in vehicle ad-hoc networks," in *Proc. ICIC*, vol. 9773, 2016, pp. 11–20.

[21] 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.

[22] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology—ASIACRYPT* (Lecture Notes in Computer Science), vol. 2894. Berlin, Germany: Springer, 2003, pp. 452–473, doi: 10.1016/j.jksuci.2018.02.016.

[23] X. Huang, Y. Mu, W. Susilo, D. S. Wong, and W. Wu, "Certificateless signature revisited," in *Proc. ACISP*, 2007, pp. 308–322.

[24] S. K. H. Islam and G. P. Biswas, "Provably secure and pairing-free certificateless digital signature scheme using elliptic curve cryptography," *Int. J. Comput. Math.*, vol. 90, no. 11, pp. 2244–2258, 2013.

[25] D. He, Y. Chen, and J. Chen, "An efficient certificateless proxy signature scheme without pairing," *Math. Comput. Model.*, vol. 57, nos. 9–10, pp. 2510–2518, 2013.

[26] N. B. Gayathri, T. Gowri, R. R. V. K. Rao, and P. V. Reddy, "Efficient and secure pairing-free certificateless directed signature scheme," *J. King Saud Univ.-Comput. Inf. Sci.*, to be published, doi: 10.1016/j.jksuci.2018.02.016.
[27] A. Malip, S.-L. Ng, and Q. Li, "A certificateless anonymous authenticated announcement scheme in vehicular ad hoc networks," *Secur. Commun. Netw.*, vol. 7, no. 3, pp. 588–601, 2014.
[28] 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.
[29] 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.
[30] J. Li, H. Yuan, and Y. Zhang, "Cryptanalysis and improvement of certificateless aggregate signature with conditional privacy preserving for vehicular sensor networks," eprint, IACR, Tech. Rep., 2016. [Online]. Available: https://eprint.iacr.org/2016/692
[31] P. Kumar, S. Kumari, V. Sharma, X. Li, A. K. Sangaiah, and S. K. H. Islam, "Secure CLS and CL-AS schemes designed for VANETs," *J. Supercomput.*, to be published, doi: 10.1007/s11227-018-2312-y.
[32] M. S. Anoop. *Elliptic Curve Cryptography: An Implementation Guide*. [Online]. Available: http://www.infosecwriters.com/text_resources/pdf/Elliptic_Curve_AnnopMS.pdf
[33] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *J. Cryptol.*, vol. 13, no. 3, pp. 361–369, 2000.
[34] P. S. L. M. Barreto, H. Y. Kim, B. Lynn, and M. Scott, "Efficient algorithms for pairing-based cryptosystems," in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science), vol. 2442. Berlin, Germany: Springer, 2002, pp. 354–368.
[35] 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.
[36] S. H. Tan, S. H. Heng, and B. M. Goi, "Java implementation for pairing-based cryptosystems," in *Computational Science and Its Applications—ICCSA* (Lecture Notes in Computer Science), vol. 6019. Berlin, Germany: Springer, 2010, pp. 188–198.
[37] *MIRACL Library*. [Online]. Available: http://certivox.org/display/EXT/MIRACLs

N. B. GAYATHRI received the M.Sc. degree in mathematics and the M.Phil. degree in cryptography from Andhra University, India, where she is currently pursuing the Ph.D. degree in cryptography. She is currently the Principal Investigator of a project under the Women Scientist Scheme, Department of Science and Technology, Government of India, India. She is involved in cryptography and information security. She is a Life Member of the Indian Mathematical Society.

GOWRI THUMBUR (M'12–SM'17) received the B.Tech. degree in electronics and communication engineering from Nagarjuna University, the M. Tech degree from Jawaharlal Nehru Technological University at Anantapur, India, and the Ph.D. degree from Jawaharlal Nehru Technological University at Kakinada, India. She is currently with the Department of Electronics and Communication Engineering, Gandhi Institute of Technology and Management, Visakhapatnam, India. Her research interests include signal processing, digital information systems and computer electronics, digital image processing, and information security. She is a Life Member of ISSS.

P. VASUDEVA REDDY received the M.Sc. and Ph.D. degrees in cryptography from S. V. University and the M. Tech. degree in CST networks from Andhra University, India. He is currently a Professor with the Department of Engineering Mathematics, Andhra University, Visakhapatnam, India. He has several publications in reputed national and international journals. His field of interest includes algebra, number theory applications, and cryptography. He is a Life Member of the Indian Mathematical Society and the Cryptology Research Society of India. He is a reviewer and advisory board member of various journals.

MUHAMMAD ZIA UR RAHMAN (M'09–SM'16) received the M.Tech. and Ph.D. degrees from Andhra University, Visakhapatnam, India. He is currently a Professor with the Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Guntur, India. He has authored and coauthored over 90 papers in international journals and proceedings. His research interests are in adaptive signal processing, biomedical signal processing, and array signal processing. He is a reviewer for various journals published by the IEEE, Springer, Elsevier, and EURASIP.