Efficient Certificateless Conditional Privacy-Preserving Authentication for VANETs

Xiaotong Zhou, Min Luo, Pandi Vijayakumar, *Senior Member, IEEE*, Cong Peng, and Debiao He, *Member, IEEE*

***Abstract*—Vehicular Ad-hoc Network (VANET) is vital for supporting intelligent transport systems, such as traffic data sharing and cooperative processing in the modern city. However, data security and privacy are the critical factors restricting the development. To address these challenges, several certificateless conditional privacy-preserving authentication (CPPA) schemes with anonymity and traceability have been proposed. These schemes avoid complicated certificate management in the PKI framework and key escrow in the ID-based protocol. However, there still exist drawbacks such as computational complexity, high communication cost or security vulnerability. Recently, Ali *et al.* proposed an efficient certificateless CPPA (CLCPPA) scheme for VANETs, but we have found that this scheme fails to resist a signature forgery attack. To achieve a trade-off between security and efficiency, we first demonstrate the insecurity of Ali *et al.'s* protocol and then introduce a security-enhanced solution. To show the feasibility and utility of our proposal, we perform a security analysis in the security model. Moreover, we evaluate the performance via comparing it with other existing schemes. From the comparison results, we can find that our scheme is more efficient than prior state-of-art solutions, in terms of signing (improving 66.75%), the verification (improving 33.19%) and bandwidth requirement (reducing 14.75%). Therefore, our proposal is more suitable to be applied in VANETs.**

Manuscript received August 23, 2021; revised February 13, 2022; accepted April 17, 2022. Date of publication April 25, 2022; date of current version July 18, 2022. This work was supported in part by the National Natural Science Foundation of China under Grants 62172307, U21A20466, 61972294, and 61932016, in part by the Blockchain Core Technology Strategic Research Program of Ministry of Education of China under Grant 2020KJ010301, in part by the Special Project on Science and Technology Program of Hubei Provience under Grant 2020AEA013, in part by the Natural Science Foundation of Hubei Province under Grant 2020CFA052, in part by the Wuhan Municipal Science and Technology Project under Grant 2020010601012187, and in part by the Foundation of Guangxi Key Laboratory of Trusted Software under Grant kx202001. The review of this article was coordinated by Dr. Ming Li. (*Corresponding authors: Min Luo; Pandi Vijayakumar.*)

Xiaotong Zhou is with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China, and also with the Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China (e-mail: xtzhou163@163.com).

Min Luo is with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China, and also with the Shandong Provincial Key Laboratory of Computer Networks, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China (e-mail: mluo@whu.edu.cn).

Pandi Vijayakumar is with the Department of Computer Science and Engineering, University College of Engineering Tindivanam, Tindivanam 604001, India (e-mail: pvijayakumar@ieee.org).

Cong Peng is with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China (e-mail: cpeng@whu.edu.cn).

Debiao He is with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China, and also with the Shanghai Key Laboratory of Privacy-Preserving Computation, MatrixElements Technologies, Shanghai 201204, China (e-mail: hedebiao@163.com).

Digital Object Identifier 10.1109/TVT.2022.3169948

***Index Terms*—Authentication, certificateless cryptography, elliptic curve, VANETs.**

I. Introduction

VEHICULAR Ad Hoc Networks (VANETs) are self-configuring ad-hoc networks typically including vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications. This traffic status exchange system allows vehicles to share messages with nearby vehicles and road side units (RSUs). RSUs can further collect the traffic message and communicate with the backbone network for data analysis, improving drive experience, road safety and traffic management [1], [2].

A classic framework of VENETs (see Fig. 1) contains Traffic Control Center (TCC), Road-Side Union (RSU), Vehicle and Internet. The TCC is a traffic management center for vehicle registration, mobility management, and strategy implementation. Typical security components in TCC include Tracing Authority (TRA) and Key Generation Center (KGC). The RSU is a short-range communication infrastructure serving for vehicles and the TCC on the roadside. The on-board unit (OBU) is equipped in every vehicle as a transmitter to achieve wireless communication via internet.

Both V2V and V2I communications rely on a Dedicated Short-Range Communication (DSRC) [3]. By the usage of the OBU and the DSRC, each vehicle periodically broadcasts information about vehicles' statuses (e.g., location and speed) and road situations (e.g., congestion situation and weather condition) to nearby vehicles or RSUs. The backbone network (e.g., TCC) accesses these messages via the Internet and provides real-time traffic services, such as adjusting traffic lights or re-routing [4].

While VENETs have great potentials to facilitate the modern intelligent transport, mobility and openness of wireless networks make VENETs vulnerable to malicious attacks [5], [6]. The attacker can generate traffic disturbances or traffic accidents by intercepting, sniffing, modifying, replaying or deleting traffic statutes. For example, when the route is congested, the attacker modifies the traffic-related data to reduce the traffic flowing. This attack can influence drivers to change their paths or even aggravates the traffic congestion.

Message authentication [7] is one of common methods to solve the above problem, but it is more likely to cause privacy risks to driver's information (e.g., location, license plate number, and traveling route). When exchanging traffic status among vehicles and RSUs, identity information is disclosed in the broadcast channel. The attacker can easily collect information and get the historical routes of targeted vehicles. This may seriously threaten user's privacy and bring about a crippling effect, such as being adopted in criminal activities. Therefore, the anonymous service is a significant part for VANETs.

However, absolute anonymity may lead to fake and faulty messages spreading in VANETs. For instance, malicious drivers may broadcast false traffic statutes, such as GPS location, weather condition and road condition, to disturb normal communication or even execute crimes. Therefore, trusted third-parties should be capable of tracing the original identity of malicious behaviors.

To mitigate the contradiction between anonymity and traceability, the conditional privacy-preserving authentication (CPPA) [8] has been proposed in VANETs. This authentication not only guarantees the validity, authenticity and integrity of messages, but also achieves the anonymity of vehicles and the traceability of malicious behaviors.

Existing CPPA schemes are mainly categorized into PKI-based [8]–[10] and ID-based [11]–[13]. PKI-based CPPA schemes can be easily deployed in VANETs, but most of them involve certificate management issues (e.g., involving a troublesome certificate revocation list). In addition, the communication overhead is also the challenge of PKI-based schemes. While ID-based schemes can ease these issues, they still suffer from key escrow problems. That is, secret keys of vehicles will be leaked once the KGC is compromised.

To solve the above issues, certificateless CPPA (CLCPPA) has been introduced in VANETs [13]–[15]. In these schemes, the vehicle has two types of private key (i.e., partial and full) which are generated by the trusted authority (i.e., KGC) and itself, respectively. Thus, an attacker who colludes with the KGC cannot obtain the vehicle's full secret key. However, most of existing schemes involve high communication costs or complex computation overheads.

Although Cui *et al.* [16] proposed a certificateless aggregate signature with high performance, their scheme has been proved to be insecure [17]. Subsequently, several improved solutions based on Cui *et al.'s* scheme have been issued [18]–[20], but none of them can satisfy security or performance requirements in practical applications. More recently, Ali *et al.* [21] designed a novel CLCPPA protocol in VANETs. This scheme only involves the Ellipse Curve Cryptography (ECC), and hence it is more efficient than many existing schemes (e.g., [17], [22], [23]). Unfortunately, there exists a security attack that Ali *et al.*'s scheme cannot resist. Thus, we are motivated to enhance the privacy and security properties of Ali *et al.*'s scheme, but without compromising the efficiency too much.

A. Our Contributions

In this paper, we first discuss the security of Ali *et al.'s* proposal [21]. Then, we propose an improved CLCPPA scheme which can achieve a trade-off between security and efficiency. The main contributions of this paper are listed as follows.

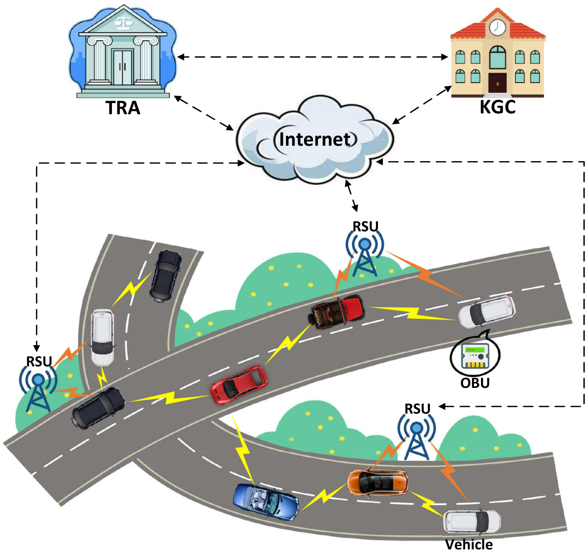

Fig. 1. Typical VANETs architecture.

- Firstly, we instantiate a digital signature forgery attack on Ali *et al.'s* scheme, which could successfully forge a message/signature pair without the targeted vehicle's secret keys. Also, we analyze possible flaws in their provable security proofs.
- Moreover, we design a new provably secure CLCPPA scheme for VANETs without any complex bilinear pairing and MapToPoint operations. Specifically, we prove our proposal can resist two types of adversaries (which will be formally defined in Section 3.3). Then, we further discuss the privacy and security properties of our scheme.
- Finally, we give the performance evaluation of our proposal and the comparison to that of other CLCPPA schemes. The result demonstrated that our proposal is more applicable for practical systems in VANETs.

B. Organization

The remainder of our article is organized below. Section II reviews existing works of CLCPPA schemes in VANETs environment. Section III describes the preliminary knowledge including network model, ECDLP and the definition of security model. Section IV reviews Ali *et al.*'s scheme together with providing possible attacks. We further introduce design details of our proposal in Section V, as well as its security analysis and performance comparison in Section VI and Section VII, respectively. The final Section VIII is the conclusion of our work.

II. Related Work

To satisfy the two main security and privacy requirements (i.e., anonymity and traceability) in VANETs, many schemes have been proposed [24]–[26].

In 2007, Raya *et al.'s* scheme [27] used the modified PKI framework to propose the first CPPA scheme for VANETs. Their scheme can ensure data authenticity, validity and integrity due to the anonymous certificate. The OBU preloads secret keys and corresponding certifications to preserve the user's real identity. During data exchanging phase, the OBU randomly selects a key pair to execute the signing algorithm. However, their scheme [27] ignored the storage cost and the communication overhead in practical applications. Numerous certificates are not only managed by the CA, but also stored and exchanged among vehicles and RSUs. In addition, the computational cost of tracing is very high.

To deal with the above problems, Lu *et al.* [28] proposed an RSU-based CPPA scheme with anonymous certificates. Under this mechanism, the vehicle can obtain temporary anonymous certificates frequently from nearby RSUs. While their scheme provides conditional privacy protection, the data sharing procedure requires RSUs to keep the online status. Other CPPA schemes based on the PKI were also introduced [13], [29], [30], but most of them confront the intractable certification management problem as mentioned above.

The ID-based CPPA is proposed for eliminating certification management problems. Neither vehicles or RSUs need to preserve certifications in the VANETs. Zhang *et al.* [31] designed an ID-based CPPA scheme, which adopted an aggregate signature function to achieve the low verification cost. However, due to the key centralized structure, existing ID-based CPPA schemes confront the key escrow issue.

As a new potential tool, CLCPPA is introduced into VANETs environment. Li *et al.* [22] proposed a certificateless CPPS protocol based on bilinear paring. In their scheme, the partial secret key of the vehicle is generated by KGC and the full secret key of the vehicle is chosen randomly by itself. However, bilinear pairing operations are time-consuming computations, which results in the inefficiency of this scheme and others such as [32] and [33].

In 2018, Cui *et al.* [16] designed a CLCPPA scheme with higher efficiency. Their scheme involved only ECC and general one-way hash function without any bilinear pairing and MapTopoint operations. However, Kamil *et al.* [17] indicated Cui *et al.'s* scheme could not withstand passive attacks. Also, Kamil *et al.* [17] presented an improvement scheme based on the previous work, but it was then proved insecure by Zhao *et al.* [20] (i.e., cannot withstand the signature forgery attack).

In 2020, Ali *et al.* [21] designed an efficient CLCPPA scheme for VANETs, which was declared provably secure in the random oracle model. Nevertheless, it seems not to provide existential unforgeability security against the potential adversary. Thus, we are motivated to propose a more secure CLCPPA scheme for VANETs but without compromising the efficiency too much.

III. Preliminary Knowledge

This section introduces a typical system architecture of VANETs, together with defining the security models and security/privacy requirements of CLCPPA scheme for VANETs.

A. System Architecture

The general system architecture for VANETs consists of four entities, i.e., TRA, KGC, RSU and Vehicle, which connect with each other via the communication channel. Our CLCPPA scheme has two levels of communications, including upper and lower levels. The upper level consists of the communications between the TRA and KGC through a secure channel, while the lower level includes V2V and V2I communications via DSRC protocol. The further detail of system entities is explained below.

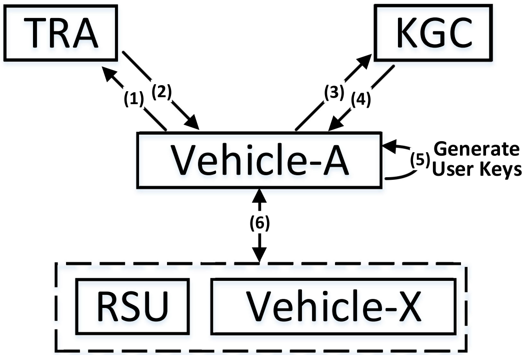

Fig. 2. The authentication steps for VANETs.

- *TRA:* The TRA is trust and with enough computing and storage abilities. This entity is responsible for setting up anonymous identities for vehicles and tracing them to the real identities (if needed).
- *KGC:* The KGC is also trust and accounting for vehicle registration and partial private key generation. In addition, the KGC owns sufficient memory and computation capability.
- *Vehicle:* The vehicle can wirelessly broadcast traffic status data to other vehicles or RSUs. Its equipped tamper-proof OBU can store private keys and system parameters. The vehicle has limited storage space, computation ability, but strong tamper-proof capacity.
- *RSU:* The RSU is a short-range communication infrastructure on the roadside. This device generally communicates with the vehicle via the DSRC protocol and verifies the received traffic-related message.

Fig. 2 shows authentication steps of CLCPPA scheme for VANETs, involving the following six steps.

1) The vehicle (e.g., Vehicle-A) applies to the TRA for registration.
2) The TRA generates anonymous identities and sends them to the vehicle. Then, the vehicle preloads them to the OBU.
3) The vehicle sends registration request to the KGC and obtains its partial private key.
4) The KGC produces and sends back the partial private key to the vehicle secretly.
5) The vehicle generates its full private key and corresponding public key for authentication.
6) The vehicle generates the signature and sends message/signature pair to surrounding vehicles or RSUs. The receivers (vehicles or RSUs) check the message.

B. Elliptic Curve Discrete Logarithm Problem

Provable security of our proposal can be reduced to the following problem.

Definition 1. Elliptic Curve Discrete Logarithm Problem (ECDLP): Define E as an elliptic curve and $\mathbb{G}$ as an additive group on E. Given $P, W \in \mathbb{G}$, the computation of $k \in Z_q^*$ is hard for any probabilistic polynomial time ($\mathcal{PPT}$) algorithms such that $W = kP$.

C. Security Model of CLCPPA

According to security requirements of the CLCPPA for VANETs [34] [35], we present two types of adversaries: Type-I ($\mathcal{A}_1$) and Type-II ($\mathcal{A}_2$). $\mathcal{A}_1$ is able to replace a vehicle's public key with a selected value, but fails to get KGC's master private key. $\mathcal{A}_2$ has the ability to get KGC's master private key, but fails to change any vehicle's public keys.

The security model of CLCPPA is defined by two types of games. In these games, a challenger $\mathcal{C}$ plays with two different adversaries $\mathcal{A}_1$ and $\mathcal{A}_2\}$. Here, we denote that $\mathcal{A}_1$ interacts with $\mathcal{C}$ in Game I and $\mathcal{A}_2$ interacts with $\mathcal{C}$ in Game II.

Definition 2: (EUF-CMA of CLCPPA) A CLCPPA scheme Γ is with existential unforgeability under adaptive chosen message attacks if no $\mathcal{PPT}$ adversary ($\mathcal{A}_1$ or $\mathcal{A}_2$) can win the following games (Game I and Game II respectively) with non-negligible probabilities.

Specifically, $\mathcal{A}$ (i.e., $\mathcal{A}_1$ or $\mathcal{A}_2$) can make following oracle queries in different games.

- *Setup:* Once $\mathcal{A}$ requests this query, $\mathcal{C}$ first generates the system parameter and master private key. Also, $\mathcal{C}$ returns the system parameter to $\mathcal{A}$.
- *H_i:* Once $\mathcal{A}$ inputs m, $\mathcal{C}$ selects an integer $z \in Z_q^*$ and adds the tuple (m, z) in the list L_{H_i}. Then, $\mathcal{C}$ outputs the result z.
- *RevealPartialSecretKey:* Once $\mathcal{A}$ requests this query with inputting AID_i, $\mathcal{C}$ sends back its partial secret key psk_i to $\mathcal{A}$.
- *RevealSecretKey:* Once $\mathcal{A}$ submits AID_i to query the secret key, $\mathcal{C}$ outputs its full secret key sk_i.
- *RevealPublicKey:* Once $\mathcal{A}$ submits AID_i to query the public key, $\mathcal{C}$ outputs its public key PK_i.
- *ReplacePublicKey:* When $\mathcal{A}$ submits the AID_i with a targeted public key PK_i', $\mathcal{C}$ updates the old public key to the targeted PK_i' in list L_k.
- *Sign:* When $\mathcal{A}$ inputs the message m_i, $\mathcal{C}$ generates and returns a signature Θ_i to $\mathcal{A}$.

Game I: The following game is executed between $\mathcal{A}_1$ and $\mathcal{C}$.

- *Initial:* $\mathcal{C}$ carries out the Setup and generates system parameters S, master secret key x and corresponding master public key P_{pub}. After that, $\mathcal{C}$ transmits S, P_{pub} to $\mathcal{A}_1$ and keeps x securely.
- *Query:* $\mathcal{A}_1$ makes queries on the oracles with any messages adaptively. The queried oracles are RevealPartialSecretKey, RevealSecretKey, RevealPublicKey, ReplacePublicKey, H_i and Sign. Assume the public key used in Sign query has been replaced, $\mathcal{A}_1$ can additionally keep the secret information corresponding to the new public key.
- *Forge:* Finally, $\mathcal{A}_1$ forges a signature Θ^* of message m^*. $\mathcal{A}_1$ wins if this forgery is valid, and $\mathcal{A}_1$ has not queried RevealPartialSecretKey, RevealSecretKey oracles and m^* in Sign oracle.

Game II: The following game is executed between $\mathcal{A}_2$ and $\mathcal{C}$.

- *Initial:* $\mathcal{C}$ carries out the Setup to produce system parameters S, master secret key x and corresponding master public key P_{pub}. After that, $\mathcal{C}$ sends S, x and P_{pub} to $\mathcal{A}_2$.
- *Query:* $\mathcal{A}_2$ can make queries on the oracles with adaptively chosen messages. The queries are RevealPartialSecretKey, RevealSecretKey, H_i and Sign. However, $\mathcal{A}_2$ cannot query ReplacePublicKey oracle in Game II.
- *Forge:* The adversary $\mathcal{A}_2$ forges a signature Θ^* of message m^*. $\mathcal{A}_2$ wins if this forgery is valid and $\mathcal{A}_2$ has not queried RevealPartialSecretKey, RevealSecretKey oracles and the signature of m^* in the query steps.

If the adversary $\mathcal{A}$ could generate a valid signature (i.e., Verify($P_{pub}, m^*, AID^*, \Theta^*$)=1) within the above conditions, we say $\mathcal{A}$ ($\mathcal{A}_1$ or $\mathcal{A}_2$) launches a successful attack in the game. If the success probability is negligible for any $\mathcal{PPT}$ adversaries, we claim that the CLCPPA scheme is EUF-CMA security.

D. Security Requirements

To achieve privacy and traceability for VANETs, the following security requirements are essential in the CLCPPA scheme [36] [37].

1) *Message authentication:* The receiver (a vehicle or an RSU) could verify the validity of the traffic status sent by a legitimate user. In addition, any modification on the traffic status will be detected.
2) *Anonymity:* A vehicle's real identity have to be transmitted anonymously and the malicious adversary cannot analyze the original identity of the message sender.
3) *Non-repudiation:* The authenticated message should be non-repudiation, meaning that no entity can deny a valid signature on the system.
4) *Conditional traceability:* The trusted third-party can get a malicious vehicle's real identity, which benefits the authority to take essential legal regulation.
5) *Un-linkability:* Only the trusted third-party can link two or more messages to the same vehicle or RSU.
6) *Resistant against attacks:* The CLCPPA scheme should withstand typical attacks existing in VANETs (e.g., replay attack, modification attack, impersonation attack).

IV. Review and Analysis of Ali *et al.'s* Proposal

This section mainly reviews Ali *et al.'s* proposal [21]. Then, we point out that this scheme fails to resist against a concrete signature forgery attack. Moreover, we analyze possible flaws in their security proof to further support the above finding.

A. Ali et al.'s Proposal

This CLCPPA scheme in [21] consists of seven algorithms, namely, Setup, GenAID, GenPSK, GenSPKGen, GenCLS, VerifyCLS and BVerifyCLS.

Setup: Given a security parameter λ, TRA and KGC initialize system parameters as follows.

1) Choose a cyclic additive group $(\mathbb{G}, q, P)$ and three secure hash functions $H_0 : \mathbb{G} \to \{0,1\}^n$, $H_1 : \mathbb{G} \times \{0,1\}^n \times \mathbb{G} \times \mathbb{G} \to \mathbb{Z}_q^*$ and $H_2 : \{0,1\} \times \mathbb{G} \times \{0,1\}^n \times \mathbb{G} \times \mathbb{G} \times \mathbb{G} \to \mathbb{Z}_q^*$.
2) TRA randomly selects $\alpha \in \mathbb{Z}_q^*$ and obtains $T_{pub} = \alpha P$.
3) KGC randomly selects $\beta \in \mathbb{Z}_q^*$ and obtains $P_{pub} = \beta P$.
4) Publish $\{\mathbb{G}, q, P, H_0, H_1, H_2, P_{pub}, T_{pub}\}$ and store master private keys α and β secretly.

GenAID: Given an original identity of vehicle RID_i, the vehicle and TRA run the algorithm together to generate an anonymous identity AID_i.

1) The vehicle randomly selects a number $r_i \in \mathbb{Z}_q^*$, calculates $AID_{i,1} = r_i P$ and sends $AID_{i,1}$ to the TRA.
2) TRA calculates $AID_{i,2} = RID_i \oplus H_0(\alpha \cdot AID_{i,1})$ and sends $(AID_{i,2}, T_i)$ to the vehicle, where T_i is a time-stamp.
3) The vehicle and TRA both store $AID_i = \{AID_{i,1}, AID_{i,2}, T_i\}$. Then, the TRA transmits AID_i to the KGC.

GenPSK: Given a vehicle's anonymous identity AID_i, KGC computes the vehicle's partial private key as follows.

1) Generate a random number $k_i \in \mathbb{Z}_q^*$.
2) Calculate $U_i = k_i P$, $\theta_i = H_1(AID_i, U_i, P_{pub})$ and $\lambda_i = (k_i + \theta_i \beta) \pmod q$.
3) Set $psk_i = \{\lambda, U_i\}$ and return it to the vehicle secretly.

GenSPK: Given a vehicle's anonymous identity AID_i, the vehicle computes the following full private key and public key.

1) Randomly generate $\mu_i \in \mathbb{Z}_q^*$ and set the private key as $sk_i = \{\mu_i, \lambda_i\}$.
2) Compute $X_i = \mu_i P_{pub}$ and $Y_i = \lambda_i P_{pub}$.
3) Set its corresponding public key $PK_i = X_i + Y_i$ and share it with surrounding vehicles and RSUs.

GenCLS: Given a message m_i, a master public key P_{pub}, an anonymous identity AID_i, a key pair $\{\mu_i, \lambda_i\}$ and PK_i, the signer runs GenCLS to compute the signature.

1) Generate a random number $a_i \in \mathbb{Z}_q^*$ and calculate $A_i = a_i P_{pub}$.
2) Compute $\delta_i = H_2(m, AID_i, PK_i, A_i, P_{pub})$ and $\eta_i = \delta_i(a_i + \mu_i + \lambda_i)$.
3) Set the signature $\Theta_i = \{\eta_i, A_i\}$. The message/signature pair is $\{m, AID_i, PK_i, \Theta_i, t_i\}$.

VerifyCLS: Given a message / signature pair $\{m, AID_i, PK_i, \Theta_i, t_i\}$ and a master public key P_{pub}, the verifier runs VerifyCLS to authenticate the message.

1) Check the freshness of T_i and t_i, and obtain $\delta_i^* = H_2(m_i, AID_i, PK_i, A_i, P_{pub})$.
2) Verify whether $\eta_i P_{pub} = \delta_i^*(A_i + PK_i)$. If not, the verifier rejects this message and accepts it otherwise.

BVerifyCLS: Given n message / signature pairs $\{m, AID_i, PK_i, \Theta_i, t_i\}_{i=1}^n$ and the master public key P_{pub}, the verifier verifies these signatures as follows.

1) Check the freshness of time-stamps T_k and t_k, respectively, where $k = 1, 2, \ldots, n$.
2) Compute $\delta_k^* = H_2(m_k, AID_k, pk_k, A_k, P_{pub})$.
3) Check whether the validity of $(\sum_{i=1}^n \eta_i) P_{pub} = \sum_{i=1}^n \delta_i(A_i + PK_i)$. If yes, the verifier accepts these signatures; otherwise, the verifier rejects them.

B. Analysis on Ali et al.'s Scheme

The proposal in [21] is claimed secure under two types of adversaries as defined above. However, this scheme fails to withstand a signature forgery attack. Specifically, assuming that a $\mathcal{PPT}$ adversary $\mathcal{A}$ is given the system master public key P_{pub}, the anonymous identity AID_i and the corresponding public key PK_i, it can forge a signature as follows.

1) Firstly, $\mathcal{A}$ randomly selects $k \in Z_q^*$ to compute $A_i' = kP_{pub} - PK_i$.
2) Then, for any message m_i, $\mathcal{A}$ calculates $\delta_i' = H_2(m_i, AID_i, PK_i, A_i', P_{pub})$ and $\eta_i' = k\delta_i'$.
3) Finally, $\mathcal{A}$ outputs (η_i', A_i') as the forged signature on m_i without using secret key.

According to $A_i' = kP_{pub} - PK_i$ and $\eta_i' = k\delta_i'$, we have:

$$\begin{aligned}\delta_i'(A_i' + PK_i) &= \delta_i'(kP_{pub} - PK_i + PK_i)\\ &= k\delta_i' \cdot P_{pub}\\ &= \eta_i' P_{pub}\end{aligned}$$

It is clear that the verification equation $\eta_i' P_{pub} = \delta_i'(A_i' + PK_i)$ is always satisfied. Therefore, $\mathcal{A}$ can forge a valid signature with only some public information (i.e., master public key, user anonymous identity and corresponding public key), without the user's secret key or system master secret key. This analysis demonstrates that Ali *et al.'s* proposal fails to resist signature forgery attacks.

In addition, the proposal in [21] cannot guarantee that the user's key pair is indeed initialized by KGC. Specifically, $\mathcal{A}$ could randomly select a fake secret key $sk_x = \{\mu_x, \lambda_x\}$ and calculates corresponding public key $PK_x = (\mu_x + \lambda_x)P_{pub}$. Then, the attacker can impersonate any valid users to pass signature verification, because the signature $\Theta_i = \{\eta_i, A_i\}$ is generated by computing $\delta_i = H_2(m_i, AID_i, PK_x, A_i, P_{pub})$ and $\eta_i = \delta_i(a_i + \mu_x + \lambda_x)$. The verification equation $\eta_i P_{pub} = \delta_i(A_i + PK_x)$ is always satisfied. Hence, we can draw the conclusion that this proposal is vulnerable to malicious attacks.

C. Analysis of Ali et al.'s Proof

This subsection further analyzes the rationality of security proof given in [21]. They claimed that the challenger $\mathcal{S}$ could correctly answer the queries from the adversary $\mathcal{F}_a^I$, but we find that the simulator cannot return the valid answer. The detail is shown as follows.

In the proof of Lemma 1 in [21], $\mathcal{F}_a^I$ makes a Signing query with message m_i. The challenger $\mathcal{S}$ selects two random numbers λ_i, σ_i from the lists. Then, $\mathcal{S}$ chooses a random number ζ_i, computes $A_i = \zeta_i P_{pub}$, sets $A_i = \lambda_i P_{pub} - X_i - Y_i$ and computes $\eta_i = \sigma_i \lambda_i \, mod \, q$. Finally, it returns signature $\Theta_i = \{\eta_i, A_i\}$. In the Forgery phase, they claimed that $\frac{\eta_i^* - \eta_i^{*'}}{\delta_i^* - \delta_i^{*'}} - (\zeta_i + \lambda_i)$ is the solution of the ECDLP.

However, there is a computational contradiction between equation $A_i = \zeta_i P_{pub}$ and equation $A_i = \lambda_i P_{pub} - X_i - Y_i$, where ζ_i and λ_i are two random numbers. Based on the ECDLP assumption, the challenger $\mathcal{S}$ cannot calculate the value of ζ_i

TABLE I
INVOLVED NOTATIONS

Notation	Description
V_i	The i-th vehicle
E	An elliptic curve
$\mathbb{G}$	An additive group
q	A prime order of $\mathbb{G}$
P	A generator of the group G
α	A secret key of the TRA
T_{pub}	A public key of the TRA
β	A secret key of the KGC
P_{pub}	A public key of the KGC
H_i	Secure hash functions
RID_i	A real identity of vehicle/RSU i
AID_i	An anonymous identity of vehicle/RSU i
psk_i	A partial secret key of vehicle/RSU i
sk_i	A full secret key of vehicle/RSU i
PK_i	A public key of vehicle/RSU i
m_i	A traffic-related message
t_i	A system timestamp
T_i	A valid time period
Θ_i	A signature of the massage

and λ_i such that $A_i = \zeta_i P_{pub}$ and $A_i = \lambda_i P_{pub} - X_i - Y_i$ in the meantime.

The proof of Lemma 2 confronts the same issue. These have demonstrated the security analysis in [21] is not adequate. Thus, Ali *et al.'s* proposal cannot be proven secure successfully.

V. Proposed Certificateless CPPA Scheme

Our proposal is composed by seven algorithms, namely, Setup, Anonymization, Extract PSK, Extract USK, Sign, Verification and BatchVerification. For convenience, notations of the proposed scheme are listed in Table I.

A. Setup

This algorithm takes a security parameter κ as an input. It first chooses a cyclic additive group ($\mathbb{G}, q, P$), and selects four secure hash functions $H_0 : \mathbb{G} \times \mathbb{G} \times \{0,1\}^n \rightarrow \{0,1\}^n$, $H_1 : \mathbb{G} \times \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{Z}_q^*$, $H_2 : \{0,1\}^n \times \mathbb{G} \times \mathbb{G} \times \mathbb{G} \times \{0,1\}^n \rightarrow \mathbb{Z}_q^*$ and $H_3 : \{0,1\}^n \times \mathbb{G} \times \mathbb{G} \times \{0,1\}^n \rightarrow \mathbb{Z}_q^*$, and then sends $\{\mathbb{G}, q, P, H_0, H_1, H_2, H_3\}$ to TRA and KGC, respectively. After that, the TRA and the KGC execute the following steps:

1) TRA randomly selects $\alpha \in \mathbb{Z}_q^*$ as master private key and calculates master public key $T_{pub} = \alpha P$.
2) KGC randomly selects $\beta \in \mathbb{Z}_q^*$ as master private key and calculates master public key $P_{pub} = \beta P$.
3) $params = \{\mathbb{G}, q, P, H_0, H_1, H_2, H_3, P_{pub}, T_{pub}\}$ is then issued as system parameter in VANETs.

B. Anonymization

This algorithm is invoked by the TRA to initialize anonymous identities of vehicles (e.g., V_i). The following is the anonymous identity generation process.

1) V_i randomly selects $r_i \in \mathbb{Z}_q^*$.
2) V_i calculates $AID_{i,1} = r_i P$ where $i = 1, 2, \ldots, n$ and sends $RID_i, AID_{i.1}$ to TRA via a secure channel.
3) TRA checks the validity of RID_i via retrieving the local database. If not, the TRA rejects this requirement; otherwise, it holds the system time-stamp T_i and calculates $AID_{i,2} = RID_i \oplus H_0(\alpha AID_{i,1}, T_{pub}, T_i)$. The anonymous identity is $AID_i = \{AID_{i,1}, AID_{i,2}, T_i\}$.
4) TRA stores AID_i and transmits it to the KGC and the vehicle. The OBU equipped in vehicle pre-loads AID_i.

C. Extract PSK

This phase is carried out by KGC to compute the partial private key of V_i. V_i sends AID_i to the KGC. Then, the KGC retrieves the AID_i from the identity list. If the AID_i exists, the KGC executes the following operations:

1) KGC randomly generates a number $k_i \in \mathbb{Z}_q^*$.
2) KGC calculates $U_i = k_i P, \theta_i = H_1(AID_i, U_i, P_{pub})$ and $\lambda_i = (k_i + \theta_i \beta) \pmod q$.
3) KGC sends the $\{\lambda_i, U_i\}$ to V_i secretly.

D. Extract USK

This phase is carries out by V_i to produce its key pair. When V_i receives $\{\lambda_i, U_i\}$, it executes as follows.

1) V_i calculates $\theta_i^* = H_1(AID_i, U_i, P_{pub})$.
2) V_i checks whether the equation $\lambda_i P = U_i + \theta_i^* P_{pub}$ holds. If not, V_i rejects the session; otherwise, V_i continues the further process.
3) V_i randomly selects $\mu_i \in \mathbb{Z}_q^*$ and then the private key is set as $sk_i = \{\mu_i, \lambda_i\}$.
4) V_i calculates $X_i = \mu_i P$ and sets its corresponding public key $PK_i = \{X_i, U_i\}$. V_i shares the public key with nearby vehicles and RSUs.

In our scheme, a batch of AID_i and $\{\lambda_i, U_i\}$ should be preloaded into the OBU. The vehicle V_i could use a unique AID_i and a partial private key $\{\lambda_i, U_i\}$ within the validity period. If the vehicle V_i runs out of all the AID_i, it will reconnect with the TRA and replenishes a stock of AID_i and $\{\lambda_i, U_i\}$ via a secure channel.

E. Sign

This algorithm is invoked by any vehicle or RSU to compute message/signature pairs. This information will be broadcasted to surrounding RSUs and vehicles. Concretely, the vehicle (e.g., V_i) executes the following operations:

1) V_i randomly selects $a_i \in \mathbb{Z}_q^*$, and calculates $A_i = a_i P, h_{1,i} = H_2(m_i, AID_i, PK_i, A_i, P_{pub}, t_i)$ and $h_{2,i} = H_3(m_i, AID_i, PK_i, A_i, P_{pub}, h_{1,i})$, where t_i is a timestamp.
2) V_i calculates $\eta_i = a_i - h_{1,i}\mu_i - h_{2,i}\lambda_i \pmod q$.
3) V_i sets the signature $\Theta_i = \{\eta_i, A_i\}$ and broadcasts $(m_i, AID_i, \theta_i, PK_i, \Theta_i, t_i)$ to surrounding vehicles or RSUs.

F. Verification

This algorithm is invoked by a verifier (vehicles or RSUs) to verify a received message/signature pair. If it is valid, the verifier can accept the message and perform further actions (e.g., re-routing) if needed. The verification process is executed below:

1) The verifier first verifies T_i and t_i. If they are not fresh, the message will be discarded.
2) The verifier checks if $\theta_i = H_1(AID_i, U_i, P_{pub})$. If it does not hold, the verifier discards this traffic status, or continues the following otherwise.
3) The verifier calculates $h^*_{1,i} = H_2(m_i, AID_i, PK_i, A_i, P_{pub}, t_i)$, $h^*_{2,i} = H_3(m_i, AID_i, PK_i, A_i, P_{pub}, h^*_{1,i})$ and $A^*_i = \eta_i P + h^*_{1,i} X_i + h^*_{2,i} U_i + (h^*_{2,i}\theta_i)P_{pub}$, respectively.
4) The verifier compares the value of A_i and A^*_i. If they are not equal, the verifier discards this message; otherwise, the verifier believe the traffic status from V_i is valid.

Proof of Correction: The equation $A_i = \eta_i P + h_{1,i} X_i + h_{2,i} U_i + (h_{2,i}\theta_i)P_{pub}$ can be verified as follows.

$$\begin{aligned} A_i &= \eta_i P + h_{1,i} X_i + h_{2,i} U_i + (h_{2,i}\theta_i)P_{pub} \\ &= [a_i - h_{1,i}\mu_i - h_{2,i}\lambda_i]P + h_{1,i} X_i + h_{2,i}(U_i + \theta_i P_{pub}) \\ &= A_i - h_{1,i} X_i - h_{2,i}(U_i + \theta_i P_{pub}) + h_{1,i} X_i + h_{2,i}(U_i \\ &\quad + \theta_i P_{pub}) \\ &= A_i. \end{aligned}$$

G. Batch Verification

This algorithm provides a batch verification of multiple message/signature pairs to improve the efficiency. When receiving $(m_i, AID_i, \theta_i, PK_i, \eta_i, t_i)^n_{i=1}$, the verifier performs the following operations.

1) The verifier verifies freshness of T_k and t_k, $\forall k = 1, 2, \ldots, n$. Those stale message/signature pairs will be discarded.
2) The verifier checks whether the equations $\theta_k = H_1(AID_k, U_k, P_{pub}), \forall k = 1, 2, \ldots, n$ holds or not. The verifier will discard those invalid messages, and continues the further verification process.
3) The verifier calculates $h^*_{1,k} = H_2(m_k, AID_k, PK_k, A_k, P_{pub}, t_i)$ and $h^*_{2,k} = H_3(m_k, AID_k, PK_k, A_k, P_{pub}, h^*_{1,k})$.
4) The verifier checks if $\sum^n_{k=1} A_k = (\sum^n_{k=1} \eta_k)P + \sum^n_{k=1}(h^*_{1,k} X_k) + \sum^n_{k=1}(h^*_{2,k} U_k) + \sum^n_{k=1}(h^*_{2,k}\theta_k)P_{pub}$. If the equation holds, the verifier accepts these messages; otherwise, the verifier rejects them.

Proof of Correction: The equation $\sum^n_{k=1} A_k = (\sum^n_{k=1} \eta_k)P + \sum^n_{k=1}(h^*_{1,k} X_k) + \sum^n_{k=1}(h^*_{2,k} U_k) + \sum^n_{k=1}(h^*_{2,k}\theta_k)P_{pub}$ holds as follows.

$$\begin{aligned} \sum^n_{k=1} A_k &= \left(\sum^n_{k=1} \eta_k\right) P + \sum^n_{k=1}(h^*_{1,k} X_k) + \sum^n_{k=1}(h^*_{2,k} U_k) \\ &+ = \sum^n_{k=1} A_k - \sum^n_{k=1}(h^*_{1,k} X_k) - \sum^n_{k=1}(h^*_{2,k} U_k) \\ &\quad - \sum^n_{k=1}(h^*_{2,k}\theta_k)P_{pub} + \sum^n_{k=1}(h^*_{1,k} X_k) \\ &\quad + \sum^n_{k=1}(h^*_{2,k} U_k) + \sum^n_{k=1}(h^*_{2,k}\theta_k)P_{pub} \\ &= \sum^n_{k=1} A_k. \end{aligned}$$

VI. Security Analysis

Our proposal can be proved to meet the aforementioned security requirements. Specifically, we employ two types of games to prove its security, which are played between a challenger $\mathcal{C}$ and two different adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$.

Theorem 1: If the ECDLP assumption holds, our proposal can be proved existentially unforgeable under Type-I adversary $\mathcal{A}_1$ in the random oracle model.

Proof: Suppose that a $\mathcal{PPT}$ adversary $\mathcal{A}_1$ can break the security of our proposal with a non-negligible probability ϵ. Then, we can construct a simulator $\mathcal{C}$ to resolve the ECDLP with ϵ^2. Specifically, $\mathcal{C}$ is given a random ECDLP instance $(P, G = x \cdot P)$, his/her goal is to solve x such that $G = x \cdot P$. In order to response to A_1's queries, $\mathcal{C}$ maintains five lists $L_{H1}, L_{H2}, L_{H3}, L_{sk}$ and L_{uk}. The concrete interactions between $\mathcal{C}$ and $\mathcal{A}_1$ are described below.

Setup: $\mathcal{C}$ generates and sends system parameters $\{\mathbb{G}, q, P, H_0, H_1, H_2, H_3, P_{pub} = G, T_{pub}\}$ to $\mathcal{A}_1$. Note that $\mathcal{A}_1$ does not know secret keys α and β.

H_1: $\mathcal{A}_1$ submits $\{AID_i, U_i, P_{pub}\}$. $\mathcal{C}$ checks the list L_{H1} with the input value. If $\{AID_i, U_i, P_{pub}\}$ exists in L_{H1}, $\mathcal{C}$ returns θ_i to $\mathcal{A}_1$; otherwise, $\mathcal{C}$ randomly chooses $\theta_i \in Z^*_q$, inserts $(AID_i, U_i, P_{pub}, \theta_i)$ to L_{H1} and finally outputs θ_i.

H_2: $\mathcal{A}_1$ submits $\{m_i, AID_i, X_i, U_i, A_i, P_{pub}, t_i\}$. $\mathcal{C}$ checks if the input value exists in the list L_{H2}, $\mathcal{C}$ returns $h_{1,i}$ to $\mathcal{A}_1$ if exists. Otherwise, $\mathcal{C}$ selects $h_{1,i} \in Z^*_q$ randomly, and then adds $\{m_i, AID_i, X_i, U_i, A_i, P_{pub}, t_i, h_{1,i}\}$ into L_{H_2}, and finally outputs $h_{1,i}$ to $\mathcal{A}_1$.

H_3: $\mathcal{A}_1$ submits $\{m_i, AID_i, X_i, U_i, A_i, P_{pub}, h_{1,i}\}$. $\mathcal{C}$ checks the list L_{H3} with the input value. If the input value already exists in the L_{H3}, $\mathcal{C}$ returns $h_{2,i}$ to $\mathcal{A}_1$; otherwise, $\mathcal{C}$ selects $h_{2,i} \in Z^*_q$ randomly, adds $\{m_i, AID_i, X_i, U_i, A_i, P_{pub}, h_{2,i}\}$ into L_{H_3} and finally outputs $h_{2,i}$ to $\mathcal{A}_1$.

RevealPartialSecretKey: $\mathcal{A}_1$ requests this query with an input AID_i. Then, $\mathcal{C}$ checks $L_{sk} = (AID_i, U_i, \lambda_i)$ as follows.

- If AID_i already exists in the L_{sk}, $\mathcal{C}$ returns $\{U_i, \lambda_i\}$ to $\mathcal{A}_1$.
- If AID_i dose not exist in the L_{sk}, $\mathcal{C}$ chooses $\lambda_i \in Z^*_q$ randomly and calculates $U_i = \lambda_i P - \theta_i P_{pub}$. Then, $\mathcal{C}$ inserts (AID_i, U_i, λ_i) to L_{sk}. Finally, $\mathcal{C}$ outputs its partial secret key $\{U_i, \lambda_i\}$ to $\mathcal{A}_1$.

RevealSecretKey: $\mathcal{A}_1$ requests this query with an input AID_i. Then, $\mathcal{C}$ checks $L_{uk} = (AID_i, U_i, \mu_i, \lambda_i, X_i)$ as follows.

- If AID_i already exists in the L_{uk}, $\mathcal{C}$ returns secret key $sk_i = \{\mu_i, \lambda_i\}$ to $\mathcal{A}_1$.
- If AID_i is not existed in the L_{uk}, $\mathcal{C}$ makes a query on RevealPartialSecretKey(AID_i) to produce (AID_i, U_i, θ_i) and adds those values to the list L_{sk}, Then, $\mathcal{C}$ chooses $\mu_i \in Z_q^*$ randomly, calculates $X_i = \mu_i P$. Finally, $\mathcal{C}$ inserts $\{AID_i, U_i, \mu_i, \lambda_i, X_i\}$ to L_{uk} and outputs the secret key $sk_i = \{\mu_i, \lambda_i\}$ to $\mathcal{A}_1$.

RevealPublicKey: $\mathcal{A}_1$ requests this query with an input AID_i. Then, $\mathcal{C}$ checks $L_{uk} = (AID_i, U_i, \mu_i, \lambda_i, X_i)$ as follows.

- If AID_i already exists in the L_{uk}, $\mathcal{C}$ returns $PK_i = \{X_i, U_i\}$ to $\mathcal{A}_1$.
- If AID_i does not exist in the L_{uk}, $\mathcal{C}$ queries RevealSecretKey(AID_i) to generate $(AID_i, U_i, \mu_i, \lambda_i, X_i)$ and adds those values to the list L_{uk}. Finally, $\mathcal{C}$ returns the public key $PK_i = \{X_i, U_i\}$ to $\mathcal{A}_1$.

ReplacePublicKey: $\mathcal{A}_1$ submits $\{AID_i, U_i'\}$, where $U_i' = k_i' P$. Then, $\mathcal{C}$ uses $\{AID_i, U_i', \mu_i, \perp, X_i\}$ to replace the old tuple value in the list L_{uk}. Note that $\mathcal{A}_1$ keeps k_i'.

Sign: $\mathcal{A}_1$ requests this query with $\{AID_i, m_i\}$. Then, $\mathcal{C}$ recovers the corresponding tuple, such as $\{AID_i, U_i, \theta_i\}$ and $\{AID_i, U_i, \mu_i, \lambda_i, X_i\}$ from L_{H_1} and L_{uk}. Then, $\mathcal{C}$ selects $\eta_i, h_{1,2}, h_{2,i} \in Z_q^*$ randomly and sets $h_{1,i} = H_2(m_i, AID_i, \ PK_i, A_i, P_{pub}, t_i)$ and $h_{2,i} = H_3(m_i, AID_i, PK_i, A_i, P_{pub}, h_{1,i})$ to the L_{H_2} and L_{H_3}, respectively. $\mathcal{C}$ calculates $A_i = \eta_i P + \delta_i(X_i + U_i + \theta_i P_{pub})$. Here, $\mathcal{C}$ returns $\Theta = (\eta_i, A_i)$ to $\mathcal{A}_1$ as a valid signature. It is easy to verify the equation $A_i = \eta_i P + h_{1,i} X_i + h_{2,i}(U_i + \theta_i P_{pub})$. Therefore, the signatures generated by $\mathcal{C}$ are indistinguishable with the actual environments.

Forge: Eventually, $\mathcal{A}_1$ returns a forged signature $\Theta = (\eta_i, A_i)$ for AID_i^* with below requirements:

- (m_i, Θ) is valid under AID_i and P_{pub}. Hence, we get:

$$\eta_i P = A_i - h_{1,i} X_i - h_{2,1}(U_i + \theta_i P_{pub}). \qquad (1)$$

- AID_i^* has not been requested in RevealSecretKey and (AID_i^*, m_i) has not been queried in Sign.

Due to the fork lemma defined in [38], the adversary A_1 can generate another valid signature $\Theta^* = (\eta_i^*, A_i)$. We get:

$$\eta_i^* P = A_i - h_{1,i} X_i - h_{2,1}^*(U_i + \theta_i P_{pub}). \qquad (2)$$

By combining two independent equations (1) and (2), we could get:

$$\begin{aligned}(\eta_i - \eta_i^*)P &= \eta_i P - \eta_i^* P \\ &= [A_i - h_{1,i} X_i - h_{2,1}(U_i + \theta_i P_{pub})] - [A_i \\ &\quad - h_{1,i} X_i - h_{2,1}^*(U_i + \theta_i P_{pub})] \\ &= h_{2,i}^*(U_i + \theta_i P_{pub}) - h_{2,i}(U_i + \theta_i P_{pub}) \\ &= (h_{2,i}^* - h_{2,i})(U_i + \theta_i P_{pub}) \\ &= (h_{2,i}^* - h_{2,i})(k_i' P + \theta_i x P) \\ &= (h_{2,i}^* - h_{2,i})(k_i' + \theta_i x)P.\end{aligned}$$

Thus, the challenger $\mathcal{C}$ outputs $[(\eta_i - \eta_i^*)(h_{2,i}^* - h_{2,i})^{-1} - k_i']\theta_i^{-1}$ as a solution of the ECDLP $(P, G = x \cdot P)$. There is not the aborted situation in the above interactions and the adopted forked lemma requires using $\mathcal{A}_1$'s ability two times. The probability of $\mathcal{C}$ solving the ECDLP is ϵ^2 (non-negligible). However, the ECDLP is actually difficult to solve for $\mathcal{PPT}$ adversaries. Thus, we can conclude that our proposal is secure against the Type-I adversary.

Theorem 2: If the ECDLP assumption holds, our proposal can be proved existentially unforgeable under Type-II adversary $\mathcal{A}_2$ in the random oracle model.

Proof: Suppose a $\mathcal{PPT}$ adversary $\mathcal{A}_2$ can break the security of our proposal with a non-negligible probability ϵ, we can construct a simulator $\mathcal{C}$ to resolve the ECDLP with $(1 - 1/q_{h1})^{2q_{ps}}(1/q_{h1})^2\epsilon^2$, where q_{h1}, q_{ps} are the time of H_1 and RevealparialSecretKey queries. Here, $\mathcal{A}_2$ is given a random ECDLP instance $(P, Q = x \cdot P)$ as an input and its goal is to compute the x. When A_2 executes oracle queries, $\mathcal{C}$ maintains five hash lists $L_{H1}, L_{H2}, L_{H3}, L_{sk}$ and L_{uk}. The random oracles queried by $\mathcal{A}_2$ are executed as follows.

Setup: $\mathcal{C}$ generates system parameters and randomly selects AID^* as a challenge anonymous identity for $\mathcal{A}_2$, Then, $\mathcal{C}$ sends system parameters $\{\mathbb{G}, q, P, H_0, H_1, H_2, H_3, P_{pub}, T_{pub}\}$ and the master secret key β to $\mathcal{A}_2$.

H_1, H_2, H_3: These oracle queries are the same as those defined in Theorem 1.

RevealPartialSecretKey: $\mathcal{A}_2$ requests this query with an input AID_i. Then, $\mathcal{C}$ checks the list $L_{sk} = (AID_i, U_i, \lambda_i)$ as follows.

- If AID_i already exists in the L_{sk}, $\mathcal{C}$ outputs $\{U_i, \lambda_i\}$ to $\mathcal{A}_2$.
- If $AID_i \neq AID^*$ and AID_i does not exist in the L_{sk}, $\mathcal{C}$ chooses $k_i \in Z_q^*$ randomly to calculate $U_i = k_i P$, and $\lambda_i = k_i + \theta_i \beta$. Then, it inserts (AID_i, U_i, λ_i) to L_{sk} and returns the partial secret key $\{U_i, \lambda_i\}$ to adversary A_i.
- If $AID_i = AID^*$, $\mathcal{C}$ aborts the game process.

RevealSecretKey: $\mathcal{A}_2$ requests this query with an input $\{AID_i\}$. $\mathcal{C}$ checks the list $L_{uk} = (AID_i, U_i, \mu_i, \lambda_i, X_i)$ as follows.

- If AID_i already exists in the L_{uk}, $\mathcal{C}$ outputs $sk_i = \{\mu_i, \lambda_i\}$ to $\mathcal{A}_2$.
- If $AID_i \neq AID^*$ and AID_i dose not exist in the L_{uk}, $\mathcal{C}$ makes a query on RevealPartialSecretKey(AID_i) to produce (AID_i, U_i, θ_i) and adds those values to the list L_{sk}. Then, $\mathcal{C}$ chooses $\mu_i \in Z_q^*$ randomly and calculates $X_i = \mu_i P$. Finally, $\mathcal{C}$ inserts $\{AID_i, U_i, \mu_i, \lambda_i, X_i\}$ to L_{uk} and outputs the secret key $sk_i = \{\mu_i, \lambda_i\}$ to $\mathcal{A}_2$.
- If $AID_i = AID^*$, $\mathcal{C}$ aborts the game process.

RevealPublicKey: $\mathcal{A}_2$ requests this query with an input AID_i. Then, $\mathcal{C}$ checks the list L_{uk} as follows.

- If AID_i already exists in the L_{uk}, $\mathcal{C}$ returns $PK_i = \{X_i, U_i\}$ to $\mathcal{A}_2$.
- If $AID_i \neq AID^*$ and AID_i does not exist in the L_{uk}, $\mathcal{C}$ queries RevealSecretKey(AID_i) to generate $(AID_i, U_i, \mu_i, \lambda_i, X_i)$. Then, $\mathcal{C}$ adds those values to L_{uk}. Finally, $\mathcal{C}$ returns the public key $PK_i = \{X_i, U_i\}$ to $\mathcal{A}_2$.

- If $AID_i = AID^*$, $\mathcal{C}$ chooses $k_i \in Z_q^*$ randomly and calculates $U_i = k_i P$. Then, $\mathcal{C}$ randomly selects a number $\theta_i \in Z_q^*$, and inserts (AID_i, U_i, θ_i) to L_{H1}. Also, $\mathcal{C}$ calculates $\lambda_i = k_i + \theta_i \beta$, sets $X_i = Q$, inserts $\{AID_i, U_i, \lambda_i\}$ to L_{sk} and $\{AID_i, U_i, \perp, \lambda_i, X_i\}$ to L_{uk} respectively. Finally, $\mathcal{C}$ returns public key $PK_i = \{X_i, U_i\}$ to $\mathcal{A}_2$.

Sign: $\mathcal{A}_2$ submits $\{AID_i, m_i\}$. Then, $\mathcal{C}$ executes as follows.

- If $AID_i \neq AID^*$, then $\mathcal{C}$ parses the corresponding tuple, such as (AID_i, U_i, θ_i) and $\{AID_i, U_i, \mu_i, \lambda_i, X_i\}$ from L_{H1} and L_{uk}, respectively. Then, $\mathcal{C}$ selects $\eta_i, \theta_i, h_{1,i}, h_{2,i} \in Z_q^*$ randomly and calculates $A_i = \eta_i P_{pub} + h_{1,i} X_i + h_{2,i}(U_i + \theta_i P_{pub})$. Here, $\mathcal{C}$ returns $\Theta = (\eta_i, A_i)$ to $\mathcal{A}_2$ and adds $h_{1,i}$ and $h_{2,i}$ to the list L_{H2} and L_{H3} respectively.
- If $AID_i = AID^*$, $\mathcal{C}$ aborts the game process.

Forge: Eventually, $\mathcal{A}_2$ returns a forged signature $\Theta = (\eta_i, A_i)$ on the message m_i for AID^*.

In forgery steps, the following requirements must satisfy:

- (m_i, Θ) is valid under the AID_i^* and the P_{pub}. Hence, we get:

$$\eta_i P = A_i - h_{1,i} X_i + h_{2,i}(U_i + \theta_i P_{pub}). \tag{3}$$

- AID_i^* has not been requested by RevealPartialSecretKey, RevealSecretKey and Sign oracle.

Due to Lemma defined in [38], A_2 forges another valid signature $\Theta^* = (\eta_i^*, A_i)$:

$$\eta_i^* P = A_i - h_{1,i}^* X_i + h_{2,i}(U_i + \theta_i P_{pub}). \tag{4}$$

By combining two independent equations (3) and (4), we could get:

$$\begin{aligned}(\eta_i - \eta_i^*)P &= \eta_i P - \eta_i^* P\\ &= [A_i - h_{1,i} X_i + h_{2,1}(U_i + \theta_i P_{pub})] -\\ &\quad [A_i - h_{1,i}^* X_i + h_{2,1}(U_i + \theta_i P_{pub})]\\ &= h_{1,i}^* X_i - h_{1,i} X_i\\ &= (h_{1,i}^* - h_{1,i}) X_i\\ &= (h_{1,i}^* - h_{1,i}) xP.\end{aligned}$$

$\mathcal{C}$ outputs $(\eta_i - \eta_i^*)(h_{1,i}^* - h_{1,i})^{-1}$ as the solution of ECDLP $(P, Q = x \cdot P)$. Here, we discuss the winning probability of $\mathcal{C}$. The probability of $\mathcal{C}$ not aborting in all queries of RevealPartialSecretKey is at most $(1 - 1/q_{h1})^{2q_{ps}}$. In addtion, $\mathcal{C}$ forging two signatures such that $AID_i = AID^*$ is greater than $(1/q_{h1})^2$. It results that $\mathcal{C}$'s advantage in solving the ECDLP is at least $(1 - 1/q_{h1})^{2q_{ps}} (1/q_{h1})^2 \epsilon^2$. This is also a contradiction. Thus, our proposal is also secure against the Type-II adversary.

Furthermore, our proposal can ensure the aforementioned types of security requirements for security and privacy protection among vehicles and RSUs.

1) *Message Authentication / Integrity:* According to existential unforgeability proofs of Theorems 1 and 2, our proposal is proven secure under Type-I and Type-II adversaries. These adversaries cannot forge a valid signature meeting $A_i = \eta_i P + h_{1,i} X_i + h_{2,i}(U_i + \theta_i P_{pub})$.
2) *Anonymity:* The anonymous identity $AID_i = \{AID_{i,1}, AID_{i,2}, T_i\}$ is used to hide the real identity, where $AID_{i,1} = r_i P$ and $AID_{i,2} = RID_i \oplus H_0(\alpha AID_{i,1}, T_{pub}, T_i)$. To extract real identity RID_i, the attacker has to compute $\alpha AID_{i,1}$. However, α is a secret key chosen by the TRA. According to the hardness of the ECDLP, there is no PPT attacker can extract α from T_{pub}. Hence, the anonymity is ensured in our proposal.
3) *Non-Repudiation:* In our proposed scheme, the signer is unable to deny a signature that has been produced previously. Assuming that the verifier attempts to deny a valid signature/message pair, the TRA could trace the real identity RID_i through its anonymous identity AID_i. Therefore, no entity can deny the validity of the signature.
4) *Conditional Traceability:* While the original identity of the vehicle RID_i is hided with AID_i, the TRA can analyze the real identity RID_i if needed. The real identity RID_i can be recovered by the TRA through using TRA's master secret key α. According to $AID_{i,1} = r_i P$ (r_i is a random number), $AID_{i,2} = RID_i \oplus H_0(\alpha AID_{i,1}, T_{pub}, T_i)$ and $AID_i = \{AID_{i,1}, AID_{i,2}, T_i\}$, the extraction process is described below:

$$\begin{aligned}RID_i &= AID_{i,2} \oplus H_0(\alpha AID_{i,1}, T_{pub}, T_i)\\ &= [RID_i \oplus H_0(\alpha AID_{i,1}, T_{pub}, T_i)] \oplus\\ &\quad H_0(\alpha AID_{i,1}, T_{pub}, T_i)\\ &= RID_i.\end{aligned}$$

   However, any $\mathcal{PPT}$ adversary cannot compute $\alpha AID_{i,1}$ due to he/she does no have the master secret key α. Therefore, only the trusted authority TRA can trace the vehicle's original identity in our proposed scheme.
5) *Un-Linkability:* In the Anonymization phase, the TRA randomly picks r_i to generate an anonymous identity. In addition, the vehicle randomly selects a_i to generate the signature. Because of the randomness of r_i and a_i, neither different anonymous identities nor different signatures of the same vehicle, can be linked by a $\mathcal{PPT}$ adversary.
6) *Resistant against attacks:* In addition to the above security properties, our proposal can also resist against the following common attacks.

- *Impersonation attack:* To launch an impersonation attack, the attacker should generate a message / signature pair $\{m_i, AID_i, \theta_i, PK_i, \Theta_i, t_i\}$ satisfying the following equations: $\theta_i = H_1(AID_i, U_i, P_{pub})$ and $A_i = \eta_i P + h_{1,i} X_i + h_{2,i}(U_i + \theta_i P_{pub})$. However, according to Theorem 1 and Theorem 2, $\mathcal{PPT}$ attackers cannot generate the valid message/signature pair, because none of them can solve the ECDLP. Thus, our proposal can successfully resist impersonation attacks.
- *Modification attack:* According to the above security analysis about our proposal, any modification of the message/signature pair $\{m_i, AID_i, \theta_i, PK_i, \Theta_i\}$ could be found by checking whether the equations $\theta_i = H_1(AID_i, U_i, P_{pub})$ and $A_i = \eta_i P + h_{1,i} X_i + h_{2,i}(U_i +$

TABLE II
NOTATIONS AND EXECUTION TIME (MS)

Notation	Description	Execution Time
T_{bp}	A bilinear pairing operation $e(P, Q)$, where $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$	4.2110
T_{bp-a}	An ECC scale multiplication operation $k \cdot P$, where $k \in Z_q^*$ and $P \in \mathbb{G}_1$	0.0071
T_{bp-m}	An ECC operation $Q = kP$, where $k \in Z_q^*$ and $P, Q \in \mathbb{G}_1$	1.7390
T_{mtp}	A map-to-point hash function	4.406
T_{ec_a}	An ECC point addition operation $P + Q$, where $P, Q \in \mathbb{G}$	0.0018
T_{ec_m}	An ECC scale multiplication operation $k \cdot P$, where $k \in Z_q^*$ and $P \in \mathbb{G}$	0.4420
T_h	A secure hash function	0.0001

$\theta_i P_{pub}$) hold or not. Hence, our proposal can also resist this attack.

- *Man-in-the-middle attack:* Our proposal achieves the message authentication feature based on the hardness of the ECDLP, so no third-party can forge valid signatures transmitted between message signers and verifiers. Hence, our proposal owns the ability to withstand this attack.
- *Replay attack:* There are two timestamps in our scheme. The t_i is contained in the message/signature pair $\{m_i, AID_i, \theta_i, PK_i, \Theta_i, t_i\}$ and the T_i is included in the anonymous identity $AID_i = \{AID_{i,1}, AID_{i,2}, T_i\}$. When receiving the message/signature pair, the verifier could detect the playback of the message by verifying the freshness of timestamps.

VII. PERFORMANCE EVALUATION

To show the feasibility of our proposal, we make a comparison of our scheme to recent CLCPPA schemes. This section discusses the efficiency of our proposal according to computation cost, communication overhead and power consumption.

A. Computation Cost

To analyze the computation cost of CLCPPA scheme, we employ the evaluation method for VANETs designed by He et al. [39].

Specifically, we mainly force on the time-consuming operation. For bilinear pairing-based schemes, the bilinear pairing is instanced as $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$, where $\mathbb{G}_1$ is an additive group of prime order q on the $E : y^2 = x^3 + x \pmod p$. Moreover, q is a 160-bit Solinas prime number and p is a 512-bit prime number. For ECC-based schemes, the ECC is instanced as an additive group G of prime order q on the $E : y^2 = x^3 + ax + b \pmod p$, where $a, b \in Z_p^*$ and p, q are both 160-bit primes.

Then, we experiment on a laptop with the MIRACL cryptographic library. The test laptop is configured with Windows 7 operation system, Intel I7-4770 3.4 GHz processor and 4 GB memory. Here, we list the execution time of involved cryptographic operations in Table II.

Next, we just demonstrate the computation measure of Mei *et al.'s* proposal [33] and our proposal. The evaluation of other related schemes can be executed by the same way. Computation costs of signature generation, single verification and batch verification are shown in Table III.

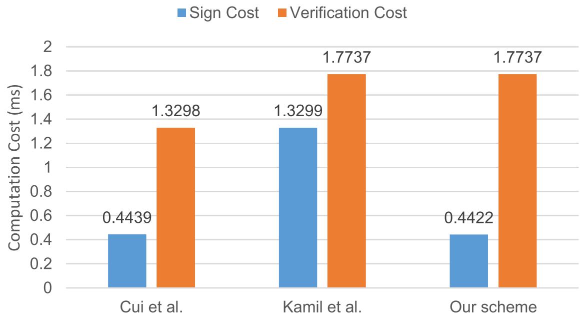

Fig. 3. The computation cost comparison (ms).

Mei *et al.*'s scheme [33] involves 4 scalar multiplication operations in $\mathbb{G}_1$, 2 point addition operations in $\mathbb{G}_1$, 2 hash-to-point operations and a general hash function operation to generate a signature. Therefore, the cost of a sign is $4T_{bp-m} + 2T_{bp-a} + 2T_{mtp} + T_h = 15.6622$ ms. To check the validity of a single message/signature pair, their proposal requires executing 4 bilinear pairing operations, 2 scalar multiplication operations in $\mathbb{G}_1$ and a general hash function operations. Therefore, the total cost of single message verification is $4T_{bp} + 2T_{bp-m} + T_h = 20.2621$ ms. If we execute batch verification algorithm, the cost of execution is $4T_{bp} + 2nT_{bp-m} + (2n - 2)T_{bp-a} + nT_h = (20.2763n - 0.0142)$ ms.

Since our proposal is pairing-free, the vehicle only needs a scalar multiplication operation in $\mathbb{G}$ and 2 general hash function operations for signing. The executing time of generation signature is $T_{ec-m} + 2T_h \approx 0.4422$ ms. To verify the individual signature, our proposal requires to run 4 scalar multiplication operations in $\mathbb{G}$, 3 point addition operations and 3 general hash function operations. The total cost time of single signature verification is $4T_{ec-m} + 3T_{ec-a} + 3T_h \approx 1.7737$ ms. For aggregate verification of n signatures, the execution time is $(2n + 2)T_{ec-m} + (3n)T_{ec-a} + (3n)T_h \approx 1.3317n + 0.442$ ms.

The percentage improvements of our proposed scheme over Kamil *et al.'s* scheme [17] are about $\frac{1.3299 - 0.4421}{1.3299} \approx 66.75\%$ and $\frac{1.3317 - 0.8897}{1.3317} \approx 33.19\%$, in terms of signature generation and batch verification.

The time cost of other proposals can be calculated in the similar way. In addition, we compare the execution cost of signature generation and single message verification of those CLCPPA schemes via a bar graph in Fig. 3. The running time of the batch verification is represented graphically in Fig. 4.

According to the comparative results (see Table III), Figs. 3 and 4. We demonstrate that our proposal requires lower communication cost than other two recent schemes [33] and [17]. While our proposal does not present much computation improvement with Cui *et al.*'s scheme [16], their scheme cannot resist Type-I and Type-II attackers mentioned in [17]. One can find that our proposal meets necessary security and privacy requirements from security analysis. Therefore, our proposal

TABLE III
THE COMPARISON OF COMPUTATION COST

Scheme	Cost of signature	Cost of single verification	Cost of aggregate verification	Secure
Mei et al. [33]	$4T_{bp-m}+2T_{bp-a}+2T_{mtp}+T_h \approx 15.6622$	$4T_{bp} + 2T_{bp-m} + T_h \approx 20.2621$	$4T_{bp} + 2nT_{bp-m} + (2n-2)T_{bp-a} + nT_h \approx 20.2763n - 0.0142$	Yes
Cui et al. [16]	$T_{ec-m} + T_{ec-a} + T_h \approx 0.4439$	$3T_{ec-m} + 2T_{ec-a} + 2T_h \approx 1.3298$	$(n+2)T_{ec-m}+(2n+2)T_{ec-a}+2nT_h \approx 0.4458n + 0.8876$	No
Kamil et al. [17]	$3T_{ec-m} + 2T_{ec-a} + 3T_h \approx 1.3299$	$4T_{ec-m} + 3T_{ec-a} + 3T_h \approx 1.7737$	$(3n+1)T_{ec-m} + 3nT_{ec-a} + 3nT_h \approx 1.3317n + 0.442$	Yes
Our Scheme	$T_{ec-m} + 2T_h \approx 0.4422$	$4T_{ec-m} + 3T_{ec_a} + 3T_h \approx 1.7737$	$(2n+2)T_{ec-m} + 3nT_{ec-a} + 3nT_h \approx 0.8897n + 0.884$	Yes

* Note that the verification phase of Kamil *et al.'s* scheme [17] omits to check R_k. Here, we recalculate its computation cost in the Table.

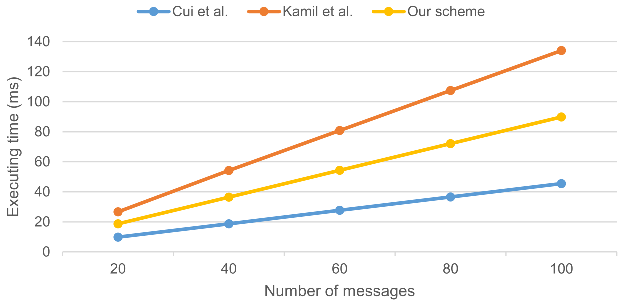

Fig. 4. Execution Time of Batch Verification.

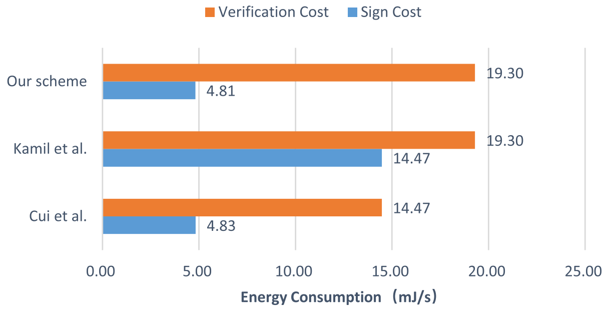

Fig. 5. Power Consumption for signature generation and message verification.

TABLE IV
COMMUNICATION OVERHEAD COMPARISON (BYTES)

Scheme	Single message	Multiple messages
Mei et al. [33]	540	540n
Cui et al. [16]	184	184n
Kamil et al. [17]	244	244n
Our scheme	208	208n

can achieve a critical trade-off between security and computation costs, and hence it is more suitable for practical applications in VANETs.

B. Communication Overhead

We further evaluate the communication overhead of our proposal and other existing CLCPPA schemes for VANETs. To analyze bilinear pairing-based schemes, we assume that the size of $|\mathbb{G}_1|$ is $64 \times 2 = 128$ bytes. For ECC-based schemes, the size of $|\mathbb{G}|$ is $20 \times 2 = 40$ bytes. In both of these schemes, the size of hashing value and a timestamp are 20 bytes and 4 bytes, respectively. Table IV shows the comparison result of communication overhead.

Mei *et al.*'s protocol involves the communication of authenticated message $\{vepk_i, PID_{i,1,j}, PID_{i,2,j}, TP, t_i, U_i, T_i\}$, where $vepk_i, PID_{i,1,j}, U_i, T_i \in \mathbb{G}_1$, $PID_{i,2,j} \in Z_q^*$ and (TP, t_i) are timestamps. Thus, the total communication overhead is $128 \times 4 + 20 + 4 \times 2 = 540$ bytes. In Cui. et al. [16], the authenticated transmitted message is $\{ID_i, vPK_i, Q_{ID_i}, R_i, S_i, t_i\}$, where $ID_i, vPK_i, Q_{ID_i}, R_i \in \mathbb{G}, S_i \in Z_q^*$ and t_i is a timestamp. Hence, the total communication overhead is $40 \times 4 + 20 + 4 = 184$ bytes.

In the verification step of Kamil. *et al.* [17], it omits to verify the validity of R_k, so the actual transmitted message is $\{PID_{y,k}, PK_k, \omega_k, R_k, \upsilon_k, T_k, \nabla, A_k, \Omega_k\}$, where $PK_k, R_k, A_k, \Omega_k \in \mathbb{G}$, $PID_{y,k}, \omega_k, \upsilon_k, \nabla \in Z_q^*$ and T_k is a timestamp. Thus, the communication overhead is $40 \times 4 + 20 \times 4 + 4 = 244$ bytes. In our proposal, the transmitted message is $\{AID_{i,1}, AID_{i,2}, T_i, X_i, U_i, \eta_i, A_i, t_i\}$, where $AID_{i,1}, X_i, U_i, A_i \in \mathbb{G}, AID_{i,2}, \eta_i \in Z_q^*$ and (T_i, t_i) are timestamps. Thus, the total communication overhead is $40 \times 4 + 20 \times 2 + 4 \times 2 = 208$ bytes.

From the above comparative results, our proposal has a lower communication overhead than [33] and [17]. Although the communication cost of our proposal is higher than that of [16], their design does not satisfy the security requirements for VANETs. Therefore, our proposed scheme can support transmitting messages among vehicles and RSUs more effectively and securely.

C. Power Consumption

On the basis of evaluation method designed by Thumbur *et al.* [40], we analyze the power utilization of our scheme. The power consumption can be computes as $E = tP$, where E is the consumed energy, t is a total time cost and P is the maximum power of CPU (10.88 W).

Table 5 shows the power consumption of the signature generation and the message verification. We can observe that our scheme can achieve a trade-off between security and power utilization. Our scheme requires lower energy than [17]. While the cost of signature verification of our scheme is higher than that of [16], their scheme is vulnerable to withstand attacks.

VIII. Conclusion

To achieve the anonymity of the vehicle and the traceability of illegal behaviors in VANETs, many CLCPPA protocols have been proposed. However, most of them are vulnerable to various attacks (e.g., the impersonation attack) or with low efficiency (e.g., involving bilinear pairing operatios). To reach a trade-off between security and efficiency, we focus on the recently efficient but insecure solution proposed by Ali et al. Specially, we first analyze the vulnerability of Ali et al's proposal and demonstrate a signature forgery attack on their scheme. Then, we design an improved CLCPPA scheme for VANETs. The security analysis presents that our proposal is with existential unforgeability against two types of adversaries in random oracle model and satisfies the necessary security requirements. In addition, the evaluation result further supports the feasibility of our scheme. Therefore, our proposal is more suitable to be applied in VANETs environment. In the future, we will continue our effort to analyze the influence of novel attacks on existing protocols, design new schemes to gain better security and efficiency and implement them in the real-word VANETs.

References

[1] M. Gonzalez-Martin, M. Sepulcre, R. Molina-Masegosa, and J. Gozálvez, "Analytical models of the performance of C-V2X mode 4 vehicular communications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1155–1166, Feb. 2019.

[2] G. Zhao, Q. Jiang, X. Huang, X. Ma, Y. Tian, and J. Ma, "Secure and usable handshake based pairing for wrist-worn smart devices on different users," *Mobile Netw. Appl.*, vol. 26, pp. 1–16, 2021.

[3] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, Jul. 2011.

[4] A. Boukerche, H. A. B. F. de Oliveira, E. F. Nakamura, and A. A. F. Loureiro, "Vehicular ad hoc networks: A new challenge for localization-based systems," *Comput. Commun.*, vol. 31, no. 12, pp. 2838–2849, 2008.

[5] 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.

[6] 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, 2015.

[7] F. Wei, P. Vijayakumar, N. Kumar, R. Zhang, and Q. Cheng, "Privacy-preserving implicit authentication protocol using cosine similarity for Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5599–5606, Apr. 2021.

[8] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "Ecpp: Efficient conditional privacy preservation protocol for secure vehicular communications," in *Proc. IEEE INFOCOM 27th Conf. Comput. Commun.*, 2008, pp. 1229–1237.

[9] M. Raya and J.-P. Hubaux, "The security of vehicular ad hoc networks," in *Proc. 3rd ACM workshop Secur. Ad Hoc Sensor Netw.*, 2005, pp. 11–21.

[10] 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.

[11] D. Huang, S. Misra, M. Verma, and G. Xue, "PACP: An efficient pseudonymous authentication-based conditional privacy protocol for VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 736–746, Sep. 2011.

[12] T. W. Chim, S. 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.

[13] M. Azees, P. Vijayakumar, and L. J. Deborah, "EAAP: Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2467–2476, Sep. 2017.

[14] H. Du and Q. Wen, "Efficient and provably-secure certificateless short signature scheme from bilinear pairings," *Comput. Stand. Interfaces*, vol. 31, no. 2, pp. 390–394, 2009.

[15] C. Fan, R. Hsu, and P. Ho, "Truly non-repudiation certificateless short signature scheme from bilinear pairings," *J. Inf. Sci. Eng.*, vol. 27, no. 3, pp. 969–982, 2011.

[16] J. Cui, J. Zhang, H. Zhong, R. Shi, and Y. Xu, "An efficient certificateless aggregate signature without pairings for vehicular ad hoc networks," *Inf. Sci.*, vol. 451-452, pp. 1–15, 2018.

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

[18] X. Hu, W. Tan, C. Yu, C. Ma, and H. Xu, "Security anlysis of certificateless aggregate signature scheme in VANETs," in *Proc. IEEE 12th Int. Congr. Image Signal Process., BioMedical Eng. Informat., CISP-BMEI 2019*, Suzhou, China, Oct. 2019, pp. 1–6.

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

[20] 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, no. 5, pp. 1–20, 2020.

[21] I. Ali, Y. Chen, N. Ullah, R. Kumar, and W. He, "An efficient and provably secure ecc-based conditional privacy-preserving authentication for vehicle-to-vehicle communication in VANETs," *IEEE Trans. Veh. Technol.*, vol. 70, no. 2, pp. 1278–1291, Feb. 2021.

[22] J. Li, H. Yuan, and Y. Zhang, "Cryptanalysis and improvement of certificateless aggregate signature with conditional privacy-preserving for vehicular sensor networks," *Networks*, vol. 317, pp. 48–66, 2015.

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

[24] F. Wei, S. Zeadally, P. Vijayakumar, N. Kumar, and D. He, "An intelligent terminal based privacy-preserving multi-modal implicit authentication protocol for internet of connected vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 3939–3951, Jul. 2021.

[25] X. Xia, S. Ji, P. Vijayakumar, J. Shen, and J. J. P. C. Rodrigues, "An efficient anonymous authentication and key agreement scheme with privacy-preserving for smart cities," *Int. J. Distrib. Sens. Netw.*, vol. 17, no. 6, 2021, Art. no. 155014772110268.

[26] H. Yang, J. Shen, T. Zhou, S. Ji, and P. Vijayakumar, "A flexible and privacy-preserving collaborative filtering scheme in cloud computing for VANETs," *ACM Trans. Internet Technol.*, vol. 22, no. 2, pp. 1–19, 2021.

[27] M. Raya and J. Hubaux, "Securing vehicular ad hoc networks," *J. Comput. Secur.*, vol. 15, no. 1, pp. 39–68, 2007.

[28] R. Lu, X. Lin, H. Zhu, P. Ho, and X. Shen, "ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications," in *Proc. INFOCOM 27th IEEE Int. Conf. Comput. Commun.*, Joint Conference of the IEEE Computer and Communications Societies, 2008, Phoenix, AZ, USA, 2008, pp. 1229–1237.

[29] A. Wasef and X. S. Shen, "EMAP: Expedite message authentication protocol for vehicular ad hoc Networks," *IEEE Trans. Mob. Comput.*, vol. 12, no. 1, pp. 78–89, Jan. 2013.

[30] M. Asghar, R. R. M. Doss, and L. Pan, "A scalable and efficient PKI based authentication protocol for VANETs," in *Proc. 28th Int. Telecommun. Netw. Appl. Conf.*, ITNAC 2018, Sydney, Australia, Nov. 2018, pp. 1–3.

[31] C. Zhang, R. Lu, X. Lin, P.-H. Ho, and X. Shen, "An efficient identity-based batch verification scheme for vehicular sensor networks," in *Proc. IEEE INFOCOM 27th Conf. Comput. Commun.*, 2008, pp. 246–250.

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

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

[34] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proc. Asiacrypt*, Taipei, Taiwan, 2003, vol. 2894, pp. 452–473.

[35] Z. Zhang, D. S. Wong, J. Xu, and D. Feng, "Certificateless public-key signature: Security model and efficient construction," *ACNS*, Berlin, Heidelberg: Springer-Verlag, LNCS, 2006, vol. 3989, pp. 293–308.

[36] S. H. Islam, M. S. Obaidat, P. Vijayakumar, E. W. Abdulhay, F. Li, and M. K. C. Reddy, “A robust and efficient password-based conditional privacy preserving authentication and group-key agreement protocol for VANETs,” *Future Gener. Comput. Syst.*, vol. 84, pp. 216–227, 2018.

[37] J. Li, Y. Ji, K. R. Choo, and D. Hogrefe, “CL-CPPA: Certificate-less conditional privacy-preserving authentication protocol for the Internet of Vehicles,” *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10332–10343, Dec. 2019.

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

[39] 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 Secur.*, vol. 10, no. 12, pp. 2681–2691, Dec. 2015.

[40] T. Gowri, G. S. Rao, P. V. Reddy, N. B. Gayathri, D. V. R. K. Reddy, and M. Padmavathamma, “Efficient and secure certificateless aggregate signature-based authentication scheme for vehicular ad hoc Networks,” *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1908–1920, Feb. 2021.

Xiaotong Zhou received the bachelor’s and master’s degrees in information security in 2012 and 2019, respectively, from Wuhan University, Wuhan, China, where she is currently working toward the Ph.D. degree with the School of Cyber Science and Engineering. Her research interests include security and privacy, including privacy protection, and blockchain security.

Min Luo received the Ph.D. degree in computer science from Wuhan University, Wuhan, China, in 2003. He is currently an Associate Professor with the School of Cyber Science and Engineering, Wuhan University. His research interests mainly include applied cryptography and blockchain technology.

Pandi Vijayakumar (Senior Member, IEEE) received the B.E. degree in computer science and engineering from Madurai Kamaraj University, Madurai, India, in 2002, the M.E. degree in computer science and engineering from the Karunya Institute of Technology, Coimbatore, India, in 2005, and the Ph.D. degree in computer science and engineering from Anna University, Chennai, India, in 2013. He is currently working as an Assistant Professor and former Dean in the Department of Computer Science and Engineering, University College of Engineering Tindivanam, Melpakkam, India, which is a constituent college of Anna University. He has 17 years of teaching experience and has produced four Ph.D. candidates successfully. He has authored or coauthored more than 100 quality papers in various IEEE transactions/journals, ACM transactions, Elsevier, IET, Springer, Wiley, and IGI Global journals. He has also authored four books for various subjects that belong to the Department of Computer Science and Engineering. He is an Associate Editor for many SCI indexed journals, namely *International Journal of Communication Systems* (Wiley), *PLOS One*, *International Journal of Semantic Web and Information Systems* (IGI Global), and *Security and Communication Networks* (Wiley|Hindawi). He is the Academic Editor of the *International Journal of Organizational and Collective Intelligence* (IGI Global), *International Journal of Software Science and Computational Intelligence* (IGI Global), *International Journal of Cloud Applications and Computing* (IGI Global), *International Journal of Digital Strategy, Governance, and Business Transformation* (IGI Global) and *Security and Privacy* (Wiley). He is also a Technical Committee Member in the Computer Communications (Elsevier) journal. He was elevated to the Editor-in-Chief Position in Cyber Security and Applications (KeAi|Elsevier) Journal. He is also listed in the world’s Top 2% Scientists for citation impact during the calendar year 2020 by Stanford University, Stanford, CA, USA.

Cong Peng received the Ph.D. degree in applied mathematics from the School of Mathematics and Statistics, Wuhan University, Wuhan, China, in 2021. He is currently an Associate Professor with the School of Cyber Science and Engineering, Wuhan University. His research interests mainly include applied cryptography and data security.

Debiao He (Member, IEEE) received the Ph.D. degree in applied mathematics from the School of Mathematics and Statistics, Wuhan University, Wuhan, China, in 2009. He is currently a Professor with the School of Cyber Science and Engineering, Wuhan University. He has authored or coauthored more than 100 research papers in refereed international journals and conferences, such as IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION SECURITY AND FORENSIC, and *Usenix Security Symposium*. His main research interests include cryptography and information security, in particular, cryptographic protocols. He was the recipient of the 2018 IEEE Systems Journal Best Paper Award and 2019 IET Information Security Best Paper Award. His work has been cited more than 10000 times at Google Scholar. He is in the Editorial Board of several international journals, such as the *Journal of Information Security and Applications*, *Frontiers of Computer Science*, and *Human-centric Computing & Information Sciences*.