I. J. Mathematical Sciences and Computing, **2024, 4, 51-62**
Published Online on December 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijmsc.2024.04.05

Authentication in VANETs with Conditional Privacy-Preserving Property Using Certificateless Aggregate Signature Schemes

Parvin Rastegari*
Electrical and Computer Engineering Group, Golpayegan College of Engineering, Isfahan University of Technology, Golpayegan, 87717-67498, Iran
E-mail: p.rastegari@iut.ac.ir
*Corresponding Author

Received: 26 February, 2024; Revised: 21 May, 2024; Accepted: 27 June, 2024; Published: 08 December, 2024

Abstract: In a Vehicular Ad Hoc Network (VANET), numerous vehicles are interconnected through a wireless network to facilitate communication. The primary objective of a VANET is to enhance driver safety and comfort by enabling the exchange of traffic-related messages within the vehicular environment. These messages can include vital information such as traffic conditions, accident alerts, and road hazards. However, addressing the security challenges in VANETs is paramount to avoid serious vulnerabilities that can compromise the entire network. One of the critical security challenges is conditional privacy-preserving authentication. This requirement mandates that each vehicle must be authenticated by other vehicles or Roadside Units (RSUs) while ensuring the privacy of the vehicle's identity. Moreover, it is essential to have the capability to trace a malicious user under specific conditions, such as in the event of a security breach or misuse of the network. In this research, we conduct an in-depth cryptanalysis of a recently proposed aggregate signature scheme designed for authentication in VANETs with conditional privacy-preserving property. Our analysis identifies the existing scheme is vulnerable against a malicious Key Generation Center (KGC) attacker, in contrast to the authors' claims. To address these issues, we propose a novel, secure, and efficient authentication scheme that maintains the conditional privacy-preserving property. We evaluate our scheme and provide a formal security proof within the Random Oracle Model (ROM). In addition to enhancing security, our scheme improves efficiency by reducing the computational and communication overhead typically associated with authentication processes in VANETs. This makes our solution not only secure but also practical for real-world deployment.

Index Terms: VANETs, Certificateless Signature Scheme, Certificateless Aggregate Signature Scheme, ROM, Authentication, Privacy

1. Introduction

According to recent studies, it is expected that the number of vehicle owners in the world exceeds to two billion by 2050 [1]. Although the increase of vehicle ownership brings a lot of prosperity to the mankind's lives, it has led to many problems cause pollution, fuel waste and the difficulties of traffic management [2]. Internet of Vehicles (IoV) is one of the important revolutions in Internet of Things (IoT) which has been emerged to handle many of the mentioned problems by allowing the vehicular users to communicate traffic-related messages (such as speed, location, direction, etc) through the internet platform [3]. The IoV environment supports different communications such as vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), vehicle-to-pedestrian (V2P), vehicle-to-network (V2N), vehicle-to-cloud (V2C) and so on [4], while a vehicular ad-hoc network (VANET) only supports V2V and V2I communications via DSRC known as 802.11p standard [5, 6].

Although VANET solves a lot of problems related to the increase of the vehicles in the world, it causes new challenges in the sense of the users' security and privacy, since the users communicate via the public environment and the risk of cyber-attacks (like eavesdropping, denial of service, forgery and replay attacks) is rapidly increasing. On the other hand, the privacy of the users is an important issue in such public environment [7]. In order to handle these security issues, various security and access control protocols have been proposed in the literature [6, 8, 9, 10]. The authors in [6] introduced a new authentication protocol for VANETs that achieves both security and privacy by enabling traceability of vehicles while preserving anonymity, addressing key challenges in vehicular network security.

Reference [8] provides a thorough review of how blockchain technology can enhance the security of VANETs, highlighting its potential to address various security challenges and proposing future research directions in this domain. In [9], an efficient privacy-preserving authentication scheme for VANETs based on the Paillier cryptosystem is proposed. In [10], an efficient batch authentication scheme incorporating rule-based access control for VANETs is presented. There are many other works in the literature which involve security challenges in VANETs.

A digital signature [11] is one of the most important cryptographic schemes which can be used to satisfy some well-known security requirements i. e. authentication, integrity and non-repudiation in VANETs. Digital signatures are constructed based on the Public Key Cryptography (PKC). In a conventional Public Key Infrastructure (PKI) setting, a Certificate Authority (CA)must issue certificates for users which links public/private keys to each other. It is obvious that handling of producing, saving and sending the certificates to the users cause difficulties in a traditional PKI. In order to handle these problems, the ID-Based PKC was introduced by Shammir et al. [12], in which the private and public keys of users are produced by a Key Generation Center (KGC)from their identifier information. The main problem of this setting is the key escrow problem, as the KGC has access to all users' private keys. In order to handle the problems of a conventional PKI and the ID-Based settings simultaneously, the notion of certificateless PKC (CL-PKC) was introduced, in which a part of a private key is chosen by the user and another part is produced by the KGC [13].

Nowadays, many cryptographic protocols based on certificateless digital signatures (CLS) are being proposed in the literature [2, 14, 15, 16]. Recently, Certificateless Aggregate Signature (CLAS) schemes are widely used to design authentication protocols in VANETs, which reduces the computation cost of verifying users' signatures by batch verification [14, 15, 16, 17, 18, 19, 20, 21, 22]. Zhou et al. [14] proposed an efficient certificateless conditional privacy-preserving authentication scheme tailored for VANETs, aiming to enhance security while minimizing computational overhead. In [15], Deng et al. introduced a CLAS scheme that is lightweight and offers provable security in the standard model. The authors in [16], proposed a CLAS scheme that enhances conditional privacy and security in VANETs while being efficiently implementable within the standard model. In [17], an enhanced CLAS scheme that addresses security vulnerabilities and improves privacy preservation in VANETs is presented. The authors in [18], analyzed the vulnerabilities of an existing signature scheme and proposed an improved version that enhances both efficiency and privacy preservation for VANET applications. In [19], a new CLAS scheme is proposed that effectively counters collusion attacks to enhance the security of VANETs. The authors in [20], introduced a novel CLAS scheme designed to efficiently ensure conditional privacy in VANETs. Thumbur et al. proposed an advanced authentication scheme leveraging CLAS schemes to enhance efficiency and security in VANETs. An extensive review of existing CLAS schemes, evaluating their efficiency and security in the context of VANETs, is provided in [22].

As previously mentioned, ensuring user privacy is a critical concern in VANETs, addressed by many recent authentication protocols through the use of pseudo-IDs in place of real IDs. However, alongside the need for privacy preservation, it is equally crucial to enable the traceability of malicious users to prevent further harm. To meet these dual objectives, the concept of conditional privacy-preserving security protocols has been introduced [14, 16, 20]. In this paper, we focus on a recently proposed Privacy Preserving Certificateless (Aggregate) Signature (PP-CL(A)S) scheme designed for VANETs [16]. Our analysis reveals vulnerabilities in the scheme, particularly its susceptibility to malicious Key Generation Center (KGC) attacks, which could compromise the security and reliability of vehicular communications. To address these shortcomings, we propose an enhanced CL(A)S scheme that ensures robustness against such attacks. Our proposed scheme not only enhances security but also maintains efficiency, making it suitable for real-world deployment in VANET environments. Furthermore, we provide a rigorous security proof within the Random Oracle Model (ROM), validating the resilience and effectiveness of our proposed solution.

The continuation of this work is prepared as follows. In Section 2, some essential preliminaries are provided. In Section 3, the model of a PP-CL(A)S scheme and its security notions for VANETs are explained. In Section 4, an overview of Wang et al.'s PP-CL(A)S scheme is provided. We introduce our designed attack against Wang et al.'s PP-CL(A)S scheme, in Section 5. Afterwards, we present our new PP-CL(A)S scheme and its security proof in ROM in Section 6. In Section 7, we compare our scheme with Wang. et al.'s scheme. Section 8 provides conclusions of our research.

2. Preliminaries

2.1. Bilinear Pairing

Let G_1 be a cyclic additive group and G_2 be a cyclic multiplicative group of a prime order q and P be a generator of G_1. A bilinear pairing $e: G_1 \times G_1 \rightarrow G_2$, defined over these groups, must have the following properties:

1. Bilinearity: $e(aP, bP) = e(P, P)^{ab}$, for all $a, b \in \mathbb{Z}_q^*$.
2. Non-degeneracy: $e(P, P) \neq 1_{G_2}$.
3. Computability: There exists an efficient algorithm to calculate $e(P, P)$.

2.2. Related Complexity Assumptions

In this paper, we deal with two complexity problems, i. e. Discrete Logarithm (DL) and Computational-Diffie-Hellman (CDH) problems and the corresponding assumptions against a Probability Polynomial Time (PPT) adversaries, which are described as follows.

DL Problem: Getting $P, aP \in G_1$ (for unknown $a \in \mathbb{Z}_q^*$) as input, obtain $a \in \mathbb{Z}_q^*$. Based on DL assumption, there is not any PPT adversary who can solve this problem with a non-negligible probability.

CDH Problem: Getting $P, aP, bP \in G_1$ (for unknown $a, b \in \mathbb{Z}_q^*$) as input, obtain $abP \in G_1$. Based on CDH assumption, there is not any PPT adversary who can solve this problem with a non-negligible probability.

3. Syntax

In this section, a system model for a VANET is described, which is based on privacy-preserving certificateless aggregate signature (PP-CLAS) to provide conditional privacy preserving authentication [16]. Afterwards, the algorithms and the security requirements of a PP-CL(A)S scheme is described [15, 16].

3.1. System Model for a VANET

The model consists of the following five entities [16]:

- Key Generation Center (KGC): produces the parameters of the system and partial private keys for vehicle users.
- Trace Authority (TRA): works with KGC to generate system parameters. Moreover, TRA assigns Pseudo-IDs to the vehicle users, such their privacy can be preserved during their communications. In fact, only TRA knows the real identities of the users and can trace the malicious vehicle users, when necessary.
- On Board Units (OBUs): OBUs are installed on vehicles to provide the possibility of V2V and V2I communications for the vehicle user.
- Roadside Units (RSUs): RSUs are located along the road to support V2I communications within their coverage area via DSRC protocol. RSUs verify the signature of the vehicle users to decide whether the received messages are from authenticated users or not. If so, they accept the messages and make the corresponding decisions. Furthermore, the RSUs create an aggregate signature and send it to the Traffic Management Center (TMC) for final decisions.
- Traffic Management Center (TMC): TMC applies batch verification on received aggregate signatures from RSUs to decide whether the messages are accepted or not. Then, it makes appropriate decisions according to the received information. So, it can handle the traffic conditions.

Authentication, integrity, non-repudiation, anonymity, unlinkability, traceability and resistance against the replay attack are the basic security requirements for VANETs. Authentication, integrity and non-repudiation are guaranteed by the corresponding CL(A)S scheme used in the structure of the authentication protocol, anonymity and unlinkability are satisfied by the pseudo-IDs generated by the TRA, traceability is achieved as the TRA can trace the real identities when necessary and finally for resistance against the replay attack, proper timestamps must be used in the communicated messages [16].

3.2. PP-CLS Scheme

A PP-CLS scheme is defined by the following phases:

- **Setup:** On input λ, outputs $params, msk$ and an identity tracking key trk. The KGC and the TRA execute this algorithm, then publish $params$ and keep msk and trk secret.
- **Pseudonym Generation (Pseudo-ID-Gen):** On input $msk, trk, params$ and a real identity of a user ID_i, outputs a pseudo-ID PID_i. The TRA executes this algorithm, then sends PID_i to the user i.
- **Partial Private Key Extract (PPK-Ext):** Getting $msk, params$ and the pseudo-ID of a user PID_i as input, returns a partial private key D_i. The KGC executes this phase and transmits D_i to the user i through a secure channel.
- **Secret Value Set (SV-Set):** On input PID_i and $params$, outputs a secret value x_i. The user i executes this phase and keeps x_i secret.
- **Public Key Set (PK-Set):** On input $params$, PID_i, D_i and x_i, returns the public key of the user i, PK_i. The user i executes this algorithm and publishes PK_i.
- **Sign:** On input $params$, PID_i, D_i, x_i and a message m_i, outputs a signature σ_i on m_i. The user i executes this algorithm as the signer and sends (m_i, σ_i) to the verifier.
- **Verify:** On input (m_i, σ_i), $params$, PID_i and PK_i, outputs 1 if σ_i is valid and 0 otherwise. Everyone can execute this phase as the verifier by the signer's public key.

Two kinds of adversaries are considered in a CL-PKC setting [15]:

- A_I who is a key replacement attacker and has access to the Hash, Public Key, Partial Private Key, Replace Public Key, Secret Value and Sign oracles, in the adversarial model.
- A_{II} who is a malicious KGC and has access to the Public Key, Secret Value and Sign oracles.

A PP-CLS scheme must satisfy the unforgeability against chosen message attack (EUF-CMA), which is considered by the following games against A_I and A_{II}, respectively.

Game 1: This game is played between a challenger C and A_I, through the following steps:

- **Initialization:** On input λ, C generates $params$ and msk. Afterwards, C sends $params$ to A_I and keeps msk secret.
- **Queries:** C must reply to all A_I's queries from the Hash, Public Key, Partial Private Key, Replace Public Key, Secret Value and Sign oracles.
- **Forgery:** A_I creates a signature σ^* of a user, with the pseudo-ID PID^* and the public key PK^*, on a message m^*.

A_I wins Game 1 if: σ^* is valid, σ^* is not an output of the Sign oracle, and A_I did not make a Partial Private Key query for PID^*.

Game 2: This game is played between a challenger C and A_{II}, through the following steps:

- **Initialization:** On input λ, C generates $params$ and msk and sends them to A_{II}.
- **Queries:** C must reply to all A_{II}'s queries from the Hash, Public Key, Secret Value and Sign oracles.
- **Forgery:** A_{II} creates a signature σ^* of a user, with the pseudo-ID PID^* and the public key PK^*, on a message m^*.

A_{II} wins Game 2 if: σ^* is valid, σ^* is not an output of the Sign oracle, and A_{II} did not make a Secret Value query for PID^*.

3.3. PP-CLAS Scheme

In a PP-CLAS scheme, an entity (who can be any users) called an aggregator changes n signatures into an individual one. This scheme is defined by the following phases:

- **Setup, Pseudo-ID-Gen, PPK-Ext, SV-Set, PK-Set, Sign, Verify:** These phases are the same as those explained in Section 3.1.
- **Aggregate (AGG):** On input $M = \{m_1, m_2, \dots, m_n\}$, $(\sigma_1, \sigma_2, \dots, \sigma_n)$, $A = \{PID_1, PK_1, PID_2, PK_2, \dots, PID_n, PK_n\}$, outputs an aggregate signature σ. This phase is executed by the aggregator.
- **Aggregate Verify (AGG-Verify):** On input (σ, M, A), outputs 1 if σ is valid and 0 otherwise. Everyone can execute this algorithm as the verifier by the signers' public keys.

A PP-CLAS scheme must guarantee the EUF-CMA, against A_I and A_{II}, which is defined by the following games.

Game 3: This game is played between a challenger C and A_I, through the following steps:

- **Initialization:** This phase is like the Initialization phase of Game 1.
- **Queries:** This phase is like the Queries phase of Game 1.
- **Forgery:** A_I creates a signature σ^* of users with pseudo-IDs and the public keys $A^* = \{PID_1^*, PK_1^*, PID_2^*, PK_2^*, \dots, PID_n^*, PK_n^*\}$, on the messages $M^* = \{m_1^*, m_2^*, \dots, m_n^*\}$.

A_I wins Game 3 if: σ^* is valid, σ^* is not an output of the Sign oracle, and A_I did not make a Partial Private Key query for at least one user in A^*.

Game 4: This game is played between a challenger C and A_{II}, through the following steps:

- **Initialization:** This phase is like the Initialization phase of Game 2.
- **Queries:** This phase is like the Queries phase of Game 2.
- **Forgery:** A_{II} creates a signature σ^* of the users in A^* on the messages in M^*.

A_I wins Game 4 if: σ^* is valid, σ^* is not an output of the Sign oracle, and A_{II} did not make a Secret Value query for at least one user in A^*.

4. An Overview of Wang et al.'s Schemes

4.1. Wang et al.'s PP-CLS Scheme

The steps of Wang et al.'s PP-CLS scheme are designed as follows [16]:

- **Setup:** On input λ, the KGC and the TRA choose a cyclic additive group G_1 and a cyclic multiplicative group G_2 of a prime order $q > 2^\lambda$ and select a bilinear pairing $e: G_1 \times G_1 \to G_2$. The KGC chooses two random generators $P, Q \in G_1$, a random number $s \in_R \mathbb{Z}_q^*$ and sets $P_{pub} = sP$. The TRA picks $k \in_R \mathbb{Z}_q^*$ and assigns $K_{pub} = kP$. Then the KGC and the TRA select collision resistant hash functions $H_1: G_1 \to \{0,1\}^*$ and $H_2, H_3: \{0,1\}^* \to \mathbb{Z}_q^*$. At last, they publish $params = \{G_1, G_2, q, e, P, Q, P_{pub}, K_{pub}, H_1, H_2, H_3\}$. Note that the KGC keeps $msk = s$ and the TRA keeps $trk = k$ secret.
- **Pseudo-ID-Gen:** To preserve the privacy of the vehicle users and provide their anonymity, the TRA assigns a pseudo-ID PID_{ij} to the vehicle i, Vh_i, with the identity ID_i, once Vh_i joins the protocol for the j'th time. To this goal, firstly Vh_ipicks a random value $t_{ij} \in_R \mathbb{Z}_q^*$ and assigns $T_{ij} = t_{ij}P$. Then Vh_i transmits (ID_i, T_{ij}) to the TRA, securely. The TRA checks the validity of ID_i and sets $PID_{ij,1} = ID_i \oplus H_1(kP + T_{ij})$ and $PID_{ij} = (PID_{ij,1}, T_{ij})$. Then the TRA sends PID_{ij} to Vh_i. It is obvious that the TRA can calculate the real identity of Vh_i by the use of $trk = k$ as $ID_i = PID_{ij,1} \oplus H_1(kP + T_{ij})$, whenever tracking the user is required.
- **PPK-Ext:** For a vehicle with the pseudo-ID PID_{ij}, the KGC picks $r_i \in_R \mathbb{Z}_q^*$ and assigns $R_i = r_iP$, $k_i = H_2(PID_{ij}, R_i)$ and $d_i = r_i + k_i s \bmod q$. Then, the KGC sends $D_i = (R_i, d_i)$ to Vh_i, securely.
- **Key-Gen:** Vh_i chooses $x_i \in_R \mathbb{Z}_q^*$ as its secret value and sets $X_i = x_iP$. Then it assigns $SK_i = (d_i, x_i)$, as its full private key, and $PK_i = (R_i, X_i)$, as its public key.
- **Sign:** To sign a traffic-related message m_i, the OBU of Vh_i chooses the current timestamp TS_i and picks $u_i \in_R \mathbb{Z}_q^*$. Then OBU computes $U_i = u_iP$, $V_i = u_iQ$, $h_i = H_3(m_i, TS_i, PID_{ij}, U_i, V_i, PK_i)$ and $W_i = (d_i + h_i x_i)Q + V_i$. Finally, OBU outputs $\sigma_i = (U_i, V_i, W_i)$ and transmits $(m_i, TS_i, PK_i, PID_{ij}, \sigma_i = (U_i, V_i, W_i))$ to the RSU which is in fact the verifier. Whenever Vh_i transmits a signature, it must request the TRA for a new pseudo-ID. Thus, each pseudo-ID can be only used once.
- **Verify:** Upon receiving the tuple $(m_i, TS_i, PK_i, PID_{ij}, \sigma_i = (U_i, V_i, W_i))$, the RSU first checks TS_i for freshness. If $TS_i$is fresh, the RSU computes $k_i = H_2(PID_{ij}, R_i)$ and $h_i = H_3(m_i, TS_i, PID_{ij}, U_i, V_i, PK_i)$. Then, the RSU verifies the following equation:

$$e(W_i, P) = e(R_i + k_i P_{pub} + h_i X_i + U_i, Q). \quad (1)$$

The RSU accepts the signature if and only if Eq. (1) holds.

Remark 1: In the Sign phase of Wang et al.'s scheme, h_i is calculated as $h_i = H_3(m_i, TS_i, PID_{ij}, U_i, V_i, W_i, PK_i)$, where W_i is computed as $W_i = (d_i + h_i x_i)Q + V_i$ [16]. This is in fact a major mistake in their proposal, as W_i is needed for calculating h_i, while it can not be computed first as its calculation needs h_i, too. So, we set h_i as $h_i = H_3(m_i, TS_i, PID_{ij}, U_i, V_i, PK_i)$, instead.

4.2. Wang et al.'s PP-CLAS Scheme

The steps of Wang et al.'s CLAS scheme are as follows [16]:

- **Setup, Pseudo-ID-Gen, PPK-Ext, Key-Gen, Sign, Verify:** These algorithms are the same as those explained in Section 3.1.
- **AGG:** On input $A = \{PID_{1j}, PK_1, \dots, PID_{nj}, PK_n\}$, $M = \{m_1, TS_1, \dots, m_n, TS_n\}$ and $(\sigma_1 = (U_1, V_1, W_1), \dots \sigma_n = (U_n, V_n, W_n))$, the RSU (as the aggregator) calculates $U = \sum_{i=1}^n U_i$, $V = \sum_{i=1}^n V_i$ and $W = \sum_{i=1}^n W_i$. Then the RSU outputs $\sigma = (U, V, W)$ as the aggregate signature of the users in A on M and sends (M, A, σ) to the TMC.
- **AGG-Verify:** Upon receiving the tuple $(M, A, \sigma = (U, V, W))$, the TMC first checks the freshness of the timestamp TS_i (for $i = 1, \dots, n$). Then it obtains $k_i = H_2(PID_{ij}, R_i)$and $h_i = H_3(m_i, TS_i, PID_{ij}, U_i, V_i, PK_i)$ (for $i = 1, \dots, n$). Then, the TMC checks the following equation:

$$e(W, P) = e(\sum_{i=1}^n (R_i + k_i P_{pub} + h_i X_i) + U, Q). \quad (2)$$

The TMC accepts the signature if and only if Eq. (2) holds.

Remark 2: The AGG phase of Wang et al.'s PP-CLAS scheme has a major mistake, as in the AGG Verify algorithm, the verifier needs to compute $h_i = H_3(m_i, TS_i, PID_{ij}, U_i, V_i, PK_i)$ (for $i = 1, \dots, n$)., while he/she has only the sum of U_is and V_is, i. e. $U = \sum_{i=1}^{n} U_i$ and $V = \sum_{i=1}^{n} V_i$ [16]. So, all U_is and V_is must be sent to the verifier and the aggregate signature must be set as $\sigma = (U_1, \dots, U_n, V_1, \dots, V_n, W)$, which its length equals to $(2n+1)|G_1| + n|timestamp|$, since all timestamps must be sent to the verifier, too. Whereas Wang et al. have claimed that their PP-CLAS scheme has a length of $3|G_1| + n|timestamp|$, in Table 5 of their paper [16], which is actually wrong. In fact, Wang et al.'s PP-CLAS scheme is not efficient in the sense of the communication cost, in contrast to one of their main claims in their paper.

5. Our Cryptanalysis of Wang et al.'s Scheme

5.1. Our Attack to Wang et al.'s PP-CLS Scheme

Wang et al. claimed that their PP-CLS scheme is EUF-CMA against A_I and A_{II} based on the CDH assumption [16]. They proved their claim in a random oracle free model. However, we will show in this section that the single signature of their proposed PP-CLS scheme is vulnerable against A_{II}.

Let A_{II} be a malicious KGC who generates all the system parameters and the master secret key similar to those explained in the Setup step of the Wang et al.'s PP-CLAS scheme in Section 4.1, except Q. To generate Q, A_{II} chooses $\gamma \in_R \mathbb{Z}_q^*$ by random and sets $Q = \gamma P$. The Pseudo-ID-Gen, PPK-Ext, SV-Set and PK-Set algorithms are the same as those explained in Section 4.1. Remember that A_{II} is a malicious KGC who has access to $D_i = (R_i, d_i)$ of the user i, but does not know x_i which is produced by the user. However, by setting $Q = \gamma P$, A_{II} can forge a signature from the user i on a message m_i^* as follows:

- A_{II} picks $u_i^* \in_R \mathbb{Z}_q^*$ by random and computes $U_i^* = u_i^* P$, $V_i^* = u_i^* Q$ and $h_i^* = H_3(m_i^*, TS_i, PID_{ij}, U_i^*, V_i^*, PK_i)$.
- A_{II} calculates W_i^* as follows:

$$W_i^* = d_i Q + h_i^* \gamma X_i + V_i^*. \tag{3}$$

- A_{II} outputs $\sigma_i^* = (U_i^*, V_i^*, W_i^*)$ as a forged signature of the user i on m_i^*.

It can be checked that $\sigma_i^* = (U_i^*, V_i^*, W_i^*)$ is a valid signature on m_i^* as:

$$\begin{aligned} e(W_i^*, P) &= e(d_i Q + h_i^* \gamma X_i + V_i^*, P) \\ &= e(d_i Q + h_i^* \gamma x_i P + u_i^* Q, P) \\ &= e(d_i Q + h_i^* x_i \gamma P + u_i^* Q, P) \\ &= e(d_i Q + h_i^* x_i Q + u_i^* Q, P) \\ &= e(d_i P + h_i^* x_i P + u_i^* P, Q) \\ &= e(R_i + k_i P_{pub} + h_i^* X_i + U_i^*, Q), \end{aligned} \tag{4}$$

which satisfies Eq. (1).

5.2. Our Attack to Wang et al.'s PP-CLAS Scheme

Wang et al. claimed that their PP-CLAS scheme is EUF-CMA against both A_I and A_{II} based on the CDH assumption [16]. They proved their claim in a random oracle free model. However, we will show in this section that their PP-CLAS scheme is not robust against A_{II}.

Let A_{II} be a malicious KGC similar to that explained in Section 5.1. Then A_{II} can forge signatures $(\sigma_1^* = (U_1^*, V_1^*, W_2^*), \dots, \sigma_n^* = (U_n^*, V_n^*, W_n^*))$ from the users in set $A = \{PID_{1j}, PK_1, \dots, PID_{nj}, PK_n\}$ on messages $M^* = \{m_1^*, \dots, m_n^*\}$ such that explained in our attack in Section 5.1. Consequently, A_{II} calculates $U^* = \sum_{i=1}^{n} U_i^*$, $V^* = \sum_{i=1}^{n} V_i^*$ and $W^* = \sum_{i=1}^{n} W_i^*$ and returns $\sigma^* = (U^*, V^*, W^*)$ as the aggregate signature of the users in A on M^*.

It can be checked that $\sigma^* = (U^*, V^*, W^*)$ is a valid signature on M^* as:

$$\begin{aligned} e(W^*, P) &= e\left(\sum_{i=1}^{n} W_i^*, P\right) = e\left(\sum_{i=1}^{n} (d_i Q + h_i^* \gamma X_i + V_i^*), P\right) \\ &= e\left(\sum_{i=1}^{n} (d_i Q + h_i^* \gamma x_i P + u_i^* Q), P\right) \\ &= e\left(\sum_{i=1}^{n} (d_i Q + h_i^* x_i \gamma P + u_i^* Q), P\right) \end{aligned}$$

$$= e\left(\sum_{i=1}^{n}(d_iQ + h_i^*x_iQ + u_i^*Q)\,,P\right)$$
$$= e\left(\sum_{i=1}^{n}(d_iP + h_i^*x_iP + u_i^*P)\,,Q\right)$$
$$= e\left(\sum_{i=1}^{n}\left(R_i + k_iP_{pub} + h_i^*X_i + U_i^*\right),Q\right)$$
$$= e\left(\sum_{i=1}^{n}\left(R_i + k_iP_{pub} + h_i^*X_i\right) + \sum_{i=1}^{n}U_i^*\,,Q\right)$$
$$= e\left(\sum_{i=1}^{n}\left(R_i + k_iP_{pub} + h_i^*X_i\right) + U^*\,,Q\right), \quad (5)$$

which satisfies Eq. (2).

6. Our New PP-CLAS Scheme

In this section, we design a PP-CLAS scheme which is robust against our attack to Wang et al.'s proposal, described earlier. Then, we present a security proof of our improvement in ROM.

6.1. The Algorithms of the New PP-CLAS Scheme

The steps of our proposed PP-CLAS scheme are as follows:

- **Setup:** Given λ, the KGC and the TRA select a cyclic additive group G of a prime order $q > 2^{\lambda}$, two random generators $P, Q \in G$, three hash functions $H_1: G_1 \to \{0,1\}^*$ and $H_2, H_3: \{0,1\}^* \to \mathbb{Z}_q^*$. Then they pick $s \in_R \mathbb{Z}_q^*$ by random and assign $P_{pub} = sP$. Finally, they publish $params = \{G, q, P, Q, P_{pub}, H_1, H_2, H_3\}$ and keep $msk = s$ secret.
- **Pseudo-ID-Gen:** To preserve the privacy of the vehicle users and provide their anonymity, the TRA assigns a pseudo-ID PID_{ij} to Vh_i with the identity $ID_i \in \{0,1\}^*$, to allow it to generate its *j*th signature. To this goal, the TRA chooses $t_{ij} \in_R \mathbb{Z}_q^*$ by random, calculates $l_i = H_1\left((s + t_{ij})Q\right)$, sets $FID_{ij} = ID_i \oplus l_i$, saves (FID_{ij}, t_{ij}) in its database and sends $PID_{ij} = (FID_{ij}, t_{ij}P)$ to Vh_i. It is obvious that the TRA can calculate the real identity of Vh_i by the use of $msk = s$ as $ID_i = FID_{ij} \oplus H_1\left((s + t_{ij})Q\right)$, whenever tracking the vehicle is required.
- **PPK-Ext:** For Vh_i with the pseudo-ID PID_{ij}, the KGC picks $r_i \in_R \mathbb{Z}_q^*$ by random, sets $R_i = r_iP$, $k_i = H_2(PID_{ij}, R_i)$ and $d_i = r_i + k_is \bmod q$. Afterwards, the KGC transmits $D_i = (R_i, d_i)$ to Vh_i, securely.
- **Key-Gen:** Vh_i chooses $x_i \in_R \mathbb{Z}_q^*$ as its secret value and calculates $X_i = x_iP$. Then it considers $SK_i = (d_i, x_i)$, as its full private key, and $PK_i = (R_i, X_i)$, as its public key.
- **Sign:** To sign a traffic-related message m_i, Vh_i considers the timestamp TS_i. Then it picks $u_i \in_R \mathbb{Z}_q^*$ and sets $U_i = u_iP$, $h_i = H_3(m_i, TS_i, U_i, PK_i, PID_{ij})$ and $v_i = d_i + h_ix_i + u_i \bmod q$. Finally, Vh_i outputs $\sigma_i = (v_i, U_i)$ and transmits $\left(m_i, TS_i, PK_i, PID_{ij}, \sigma_i = (v_i, U_i)\right)$ to the RSU. Whenever Vh_i transmits a signature, it must request the TRA for a new pseudo-ID. Thus, each pseudo-ID can be only used once.
- **Verify:** Upon receiving the tuple $\left(m_i, TS_i, PK_i, PID_{ij}, \sigma_i = (v_i, U_i)\right)$, the RSU first checks TS_i. If TS_i is fresh, the RSU computes $k_i = H_2(PID_{ij}, R_i)$ and $h_i = H_3(m_i, TS_i, U_i, PK_i, PID_{ij})$. Then the RSU checks the following equation:

$$v_iP = R_i + k_iP_{pub} + h_iX_i + U_i. \quad (6)$$

  If Eq. (6) holds, the RSU outputs1. Otherwise, it outputs 0.
- **AGG:** On input $A = \{PID_{1j}, PK_1, \dots, PID_{nj}, PK_n\}$, $M = \{m_1, TS_1, \dots, m_n, TS_n\}$ and $(\sigma_1 = (v_1, U_1), \dots \sigma_n = (v_n, U_n))$, the RSU (as the aggregator) calculates $v = \sum_{i=1}^{n} v_i$. Then the RSU returns $\sigma = (v, U_1, \dots, U_n)$ as the aggregate signature of the users in A on M and sends (M, A, σ) to the TMC.
- **AGG-Verify:** Upon receiving the tuple $(M, A, \sigma = (v, U_1, \dots, U_n))$, the TMC first checks the freshness of TS_i (for $i = 1, \dots, n$). Then it calculates $k_i = H_2(PID_{ij}, R_i)$ and $h_i = H_3(m_i, TS_i, U_i, PK_i, PID_{ij})$ (for $i = 1, \dots, n$). Then, the TMC checks the equality:

$$vP = \sum_{i=1}^{n}\left(R_i + k_iP_{pub} + h_iX_i + U_i\right). \quad (7)$$

If Eq. (7) holds, the TMC outputs1. Otherwise, it returns 0.

6.2. Security Analysis of our PP-CLAS Scheme

Correctness:

The individual signature satisfies the correctness property, as:

$$\begin{aligned} v_iP &= (d_i + h_i x_i + u_i)P \\ &= (r_i + k_i s + h_i x_i + u_i)P \\ &= r_iP + k_i sP + h_i x_i P + u_i P \\ &= R_i + k_i P_{pub} + h_i X_i + U_i, \end{aligned} \tag{8}$$

which shows that Eq. (6) is correct. Moreover:

$$\begin{aligned} vP &= \sum_{i=1}^{n} v_iP = \sum_{i=1}^{n}(d_i + h_i x_i + u_i)P \\ &= \sum_{i=1}^{n}\left(R_i + k_i P_{pub} + h_i X_i + U_i\right), \end{aligned} \tag{9}$$

which shows that Eq. (7) is correct.

Unforgeability:

In this section, we prove that our proposed PP-CLAS scheme is EUF-CMA, against A_I and A_{II}, based on the DL assumption, in ROM. To this goal, we only prove the unforgeability of the individual signature, which results in the unforgeability of the aggregate signature, too.

Lemma 1. If an adversary A_I exists who can be a winner of Game 1, with a non-negligible probability, then we can arrange an algorithm C, which can resolve an instance of the DL problem with a non-negligible probability.

Proof. Assume that C takes $P, aP \in G$ as input to find $a \in \mathbb{Z}_q^*$. For beginning, C generates a list $L = \{PID_{ij}, d_i, x_i, R_i, X_i, l_i, k_i, (m_i, U_i, h_i)\}$, which is firstly empty. Then C executes Game 1 as follows:

- **Initialization:** On input λ, C sets $P_{pub} = aP$. Afterwards, it generates other system parameters similar to that described in the Setup algorithm in Section 6.1 and transmits $params = \{G, q, P, Q, P_{pub}, H_1, H_2, H_3\}$ to A_I. Note that C does not know $msk = a$.
- **Queries:** A_I issues PK, H_1, H_2, H_3, D, SK, RPK and Sign queries and C replies to them as follows:
  - PK queries: Upon receiving a PK query for the vehicle Vh_i from A_I, C acts as follows:
    - if $Vh_i \neq Vh_i^*$, C picks $d_i, x_i, k_i, t_{ij} \in_R \mathbb{Z}_q^*$ and $l_i \in_R \{0,1\}^*$, sets $FID_{ij} = ID_i \oplus l_i$, $PID_{ij} = \left(FID_{ij}, t_{ij}P\right)$, $X_i = x_iP$ and $R_i = d_iP - k_iP_{pub}$ and sends PID_{ij} and $PK_i = (R_i, X_i)$ to A_I. Moreover, C adds $\{PID_{ij}, d_i, x_i, R_i, X_i, l_i, k_i\}$ in L.
    - if $Vh_i = Vh_i^*$, C picks $r_i^*, x_i^*, t_{ij}^* \in_R \mathbb{Z}_q^*$ and $l_i^* \in_R \{0,1\}^*$, sets $FID_{ij}^* = ID_i^* \oplus l_i^*$, $PID_{ij}^* = \left(FID_{ij}^*, t_{ij}^*P\right)$, $X_i^* = x_i^*P$, $R_i^* = r_i^*P$ and $k_i^* = H_2\left(PID_{ij}^*, R_i^*\right)$. Then C sends PID_{ij}^* and $PK_i^* = (R_i^*, X_i^*)$ to A_I. Moreover, C inserts $\{PID_{ij}^*, d_i^* = \perp, x_i^*, R_i^*, X_i^*, l_i^*, k_i^*\}$ into L.
  - H_1 queries: Upon receiving a H_1 query for Vh_i from A_I, C searches in L to find l_i. If l_i is found in L, C sends it to A_I. Otherwise, C runs the PK query to obtain l_i and sends it to A_I.
  - H_2 queries: Upon receiving a H_2 query for Vh_i from A_I, C searches in L to find k_i. If k_i is found in L, C sends it to A_I. Otherwise, C runs the PK query to obtain k_i and sends it to A_I.
  - H_3 queries: Upon receiving a H_3 query for (m_i, U_i) from A_I, C searches in L to find h_i. If h_i is found in L, C sends it to A_I. Otherwise, C picks $h_i \in_R \mathbb{Z}_q^*$ by random and returns it to A_I. Furthermore, C adds (m_i, U_i, h_i) in L.
  - D queries: Upon receiving a D query for the vehicle Vh_i from A_I, C searches in L to find R_i and d_i. If R_i and d_i are found in L, C sends $D_i = (R_i, d_i)$ to A_I. Otherwise, C runs the PK query to obtain R_i and d_i and returns $D_i = (R_i, d_i)$ to A_I.
  - SK queries: Upon receiving a SK query for the vehicle Vh_i from A_I, C searches in L to find d_i and x_i. If d_i and x_i are found in L, C sends $SK_i = (x_i, d_i)$ to A_I. Otherwise, C runs the PK query to obtain d_i and x_i and returns $SK_i = (x_i, d_i)$ to A_I.
  - RPK queries: When A_I wants to replace a public key $PK_i = (R_i, X_i)$ with $PK_i' = (R_i', X_i')$, C applies this query and replaces $\{d_i, x_i, R_i, X_i\}$ with $\{\perp, \perp, R_i', X_i'\}$ in L.
  - Sign queries: Upon receiving a Sign query of Vh_i on m_i from A_I, C searches in L to find $\{PID_{ij}, d_i, x_i, R_i, X_i, l_i, k_i\}$. If these instances do not exist in L, C executes the corresponding queries to create them in L. Then C acts as follows:
    - If $x_i \neq \perp$ and $d_i \neq \perp$, C picks $u_i \in_R \mathbb{Z}_q^*$, sets $U_i = u_iP$, $h_i = H_3\left(m_i, TS_i, U_i, PK_i, PID_{ij}\right)$ and $v_i = d_i + h_i x_i + u_i \mod q$, where TS_i is the current timestamp. Then C transmits $\left(m_i, TS_i, PK_i, PID_{ij}, \sigma_i = (v_i, U_i)\right)$ to A_I. Finally, C refreshes L by inserting (m_i, U_i, h_i) in it and repeating the execution of PK query.

- If $x_i = \perp$ or $d_i = \perp$, C chooses $v_i, h_i \in_R \mathbb{Z}_q^*$, calculates $U_i = v_i P - \left(R_i + k_i P_{pub}\right) - h_i x_i$ and transmits $\left(m_i, TS_i, PK_i, PID_{ij}, \sigma_i = (v_i, U_i)\right)$ to A_I, where TS_i is the current timestamp. Then C refreshes L by inserting (m_i, U_i, h_i) in it and repeating the execution of PK query.

- **Output:** A_I forges a valid signature $\sigma_i^* = (v_i^*, U_i^*)$ of Vh_i^* on m_i^*, where $U_i^* = u_i^* P$ and:

$$v_i^* = r_i^* + k_i^* a + h_i^* x_i^* + u_i^*. \tag{10}$$

According to forking lemma [23], C can replay A_I to produce another valid signature $\hat{\sigma}_i^* = (\hat{v}_i^*, U_i^*)$, with the same random tapes, but different reply to H_2 query, i. e. $\hat{k}_i^*$. We have:

$$\hat{v}_i^* = r_i^* + \hat{k}_i^* a + h_i^* x_i^* + u_i^*. \tag{11}$$

Finally, C can solve the corresponding DL problem by obtaining a from Eq. (10) and Eq. (11), as:

$$a = \frac{v_i^* - \hat{v}_i^*}{k_i^* - \hat{k}_i^*},$$

which contradicts to the DL assumption in complexity theory.

Lemma 2. If an adversary A_{II} exists who can be a winner of Game 2, with a non-negligible probability, then we can arrange an algorithm C, which can resolve an instance of the DL problem with a non-negligible probability.

Proof. Assume that C takes $P, aP \in G$ as input to find $a \in \mathbb{Z}_q^*$. For beginning, C generates a list $L = \{PID_{ij}, d_i, x_i, R_i, X_i, l_i, k_i, (m_i, U_i, h_i)\}$, which is firstly empty. Then C executes Game 2 as follows:

- **Initialization:** On input λ, C generates all system parameters similar to that explained in the Setup algorithm in Section 6.1 and transmits $params = \{G, q, P, Q, P_{pub}, H_1, H_2, H_3\}$ and $msk = s$ to A_{II}. Note that as A_{II} is a malicious KGC, it knows $msk = s$, here.
- **Queries:** A_{II} issues PK, H_1, H_2, H_3, SK and Sign queries and C replies to them as follows:
  - PK queries: Upon receiving a PK query for the vehicle Vh_i from A_{II}, C acts as follows:
    - if $Vh_i \neq Vh_i^*$, C picks $x_i, r_i, t_{ij} \in_R \mathbb{Z}_q^*$, computes $l_i = H_1\left(\left(s + t_{ij}\right)Q\right)$, \$ sets $FID_{ij} = ID_i \oplus l_i$, $PID_{ij} = \left(FID_{ij}, t_{ij}P\right)$, $X_i = x_i P$ and $R_i = d_i P - k_i P_{pub}$ and returns PID_{ij} and $PK_i = (R_i, X_i)$ to A_{II}. Moreover, C inserts $\{PID_{ij}, d_i, x_i, R_i, X_i, l_i, k_i\}$ into L.
    - if $Vh_i = Vh_i^*$, C sets $X_i^* = aP$. Then C picks $r_i^*, t_{ij}^* \in_R \mathbb{Z}_q^*$, sets $l_i^* = H_1\left(\left(s + t_{ij}^*\right)Q\right)$, $FID_{ij}^* = ID_i^* \oplus l_i^*$, $PID_{ij}^* = \left(FID_{ij}^*, t_{ij}^* P\right)$, $R_i^* = r_i^* P$, $k_i^* = H_2\left(PID_{ij}^*, R_i^*\right)$ and $d_i^* = r_i^* + k_i^* s$ and sends PID_{ij}^* and $PK_i^* = (R_i^*, X_i^*)$ to A_{II}. Moreover, C inserts $\{PID_{ij}^*, d_i^*, x_i^* = \perp, R_i^*, X_i^*, l_i^*, k_i^*\}$ into L.
  - H_1 queries: Upon receiving a H_1 query for Vh_i from A_{II}, C searches in L to find l_i. If l_i is found in L, C sends it to A_{II}. Otherwise, C runs the PK query to obtain l_i and sends it to A_{II}.
  - H_2 queries: Upon receiving a H_2 query for Vh_i from A_{II}, C searches in L to find k_i. If k_i is found in L, C sends it to A_{II}. Otherwise, C runs the PK query to obtain k_i and sends it to A_{II}.
  - H_3 queries: Upon receiving a H_3 query for (m_i, U_i) from A_I, C searches in L to find h_i. If h_i is found in L, C sends it to A_I. Otherwise, C picks $h_i \in_R \mathbb{Z}_q^*$ by random and returns it to A_I. Furthermore, C adds (m_i, U_i, h_i) in L.
  - SK queries: Upon receiving a SK query for Vh_i from A_{II}, C searches in L to find d_i and x_i. If d_i and x_i are found in L, C returns $SK_i = (x_i, d_i)$ to A_{II}. Otherwise, C executes the PK query to create d_i and x_i in L and sends $SK_i = (x_i, d_i)$ to A_{II}.
  - Sign queries: Upon receiving a Sign query of Vh_i on m_i from A_I, C searches in L to find $\{PID_{ij}, d_i, x_i, R_i, X_i, l_i, k_i\}$. If these instances do not exist in L, C executes the corresponding queries to create them in L. Then C acts as follows:
    - If $x_i \neq \perp$, C picks $u_i \in_R \mathbb{Z}_q^*$, sets $U_i = u_i P$, $h_i = H_3\left(m_i, TS_i, U_i, PK_i, PID_{ij}\right)$ and $v_i = d_i + h_i x_i + u_i \mod q$, where TS_i is the current timestamp. Then C transmits $\left(m_i, TS_i, PK_i, PID_{ij}, \sigma_i = (v_i, U_i)\right)$ to A_{II}. Finally, C refreshes L by inserting (m_i, U_i, h_i) in it and repeating the execution of PK query.
    - If $x_i = \perp$, C chooses $v_i, h_i \in_R \mathbb{Z}_q^*$, calculates $U_i = v_i P - \left(R_i + k_i P_{pub}\right) - h_i x_i$ and transmits $\left(m_i, TS_i, PK_i, PID_{ij}, \sigma_i = (v_i, U_i)\right)$ to A_{II}, where TS_i is the current timestamp. Then C refreshes L by inserting (m_i, U_i, h_i) in it and repeating the execution of PK query.

- **Output:** A_{II} forges a valid signature $\sigma_i^* = (v_i^*, U_i^*)$ of Vh_i^* on m_i^*, where $U_i^* = u_i^* P$ and:

$$v_i^* = r_i^* + k_i^* s + h_i^* a + u_i^*. \tag{12}$$

According to forking lemma [23], C can replay A_{II} to produce another valid signature $\hat{\sigma}_i^* = (\hat{v}_i^*, U_i^*)$, with the same random tapes, but different reply to H_3 query, i. e. $\hat{h}_i^*$. We have:

$$\hat{v}_i^* = r_i^* + k_i^* s + \hat{h}_i^* a + u_i^*. \tag{13}$$

Finally, C can solve the corresponding DL problem by obtaining a from Eq. (12) and Eq. (13), as:

$$a = \frac{v_i^* - \hat{v}_i^*}{h_i^* - \hat{h}_i^*},$$

which contradicts to the DL assumption in complexity theory.

Theorem 1. Our proposed PP-CL(A)S scheme is unforgeable in the certificateless setting based on the random oracle assumption.

Proof. It is straightforward to imply Theorem 1 from Lemma 1 and Lemma 2.

Other Security Requirements:

As mentioned in Section 3.1, authentication, integrity, non-repudiation, anonymity, unlinkability, traceability and resistance against the replay attack are the basic security requirements for VANETs. Our proposed protocol satisfies the authentication, integrity and non-repudiation, since the corresponding PP-CL(A)S scheme used in its structure is unforgeable against both types of adversaries in the certificateless setting, i. e. A_I and A_{II}, based on our proof in ROM provided in Theorem 1. Anonymity and unlinkability are satisfied, as we use pseudo-IDs instead of real IDs. Moreover, the real IDs are traceable by the TRA, when necessary. Finally, as we used timestamps in communicated messages, our protocol is resistant against the replay attack.

7. Comparisons

In this section, we compare our proposed PP-CLAS scheme with Wang et al.'s PP-CLAS scheme [16]. Table 1 provides a comparison between the computation costs, while Table 2 provides a comparison between the communication costs and resistance against two kinds of well-known adversaries in the certificateless setting, i. e. A_I and A_{II}. In these tables, n denotes the number of the signer vehicles. T_P, T_M and T_A denote the required time for computing a pairing operation, a multiplication and a point addition on an elliptic curve, which are assumed to be equal to 15.0738 ms, 1.9456 ms and 0.014 ms, respectively. Moreover, T_H denotes the required time for a hash function operation, which is assumed to be 0.001 ms [16]. Furthermore, let $|G_1|$, $|G|$and $|\mathbb{Z}_q^*|$ be the size of an element in G_1, G and $\mathbb{Z}_q^*$, respectively.

As shown in Table 1 and Table 2, our proposed scheme is not only resistant against a malicious KGC A_{II}, but also it is much more efficient than Wang et al.'s scheme in both computations related and communications related costs.

Table 1. Comparison Between Computation Costs

Scheme	Sign	Verification	Aggregate Signature Verification
Wang et al.'s	$3T_M + 1T_A + 1T_H$ ($\approx 5.8518\ ms$)	$2T_P + 2T_M + 3T_A + 2T_H$ ($\approx 34.0828\ ms$)	$2T_P + 2nT_M + (2n+1)T_A + 2nT_H$ ($\approx 30.1616 + 3.9212n\ ms$)
Ours	$1T_M + 1T_H$ ($\approx 1.9466\ ms$)	$3T_M + 3T_A + 2T_H$ ($\approx 5.8808\ ms$)	$(2n+1)T_M + 3nT_A + 2nT_H$ ($\approx 31.9456 + 3.93522n\ ms$)

Table 2. Comparison Between Communication Costs and Resistance against A_I and A_{II}

Scheme	PK Size	Signature Size	Aggregate Signature Size	A_I	A_{II}
Wang et al.'s	$2\|G_1\|$	$3\|G_1\|$	$(2n+1)\|G_1\|$	✓	×
Ours	$2\|G\|$	$1\|\mathbb{Z}_q^*\| + 1\|G\|$	$1\|\mathbb{Z}_q^*\| + n\|G\|$	✓	✓

8. Conclusions

In this paper, we investigated privacy-preserving certificateless aggregate signature (PP-CLAS) schemes for developing conditional privacy-preserving authentication protocols tailored for VANETs, an area of significant recent interest. Our study involved a thorough cryptanalysis of a protocol recently proposed by Wang et al., revealing

vulnerabilities to malicious Key Generation Centers (KGC) despite their claims of robustness. To address these shortcomings, we introduced a novel scheme designed to overcome the identified weaknesses in Wang et al.'s approach. Subsequently, we provided a rigorous security proof of our scheme within the random oracle model (ROM). Importantly, our analysis demonstrated that our proposed scheme not only achieves resilience against malicious KGCs but also exhibits superior efficiency compared to the protocol by Wang et al. These findings underscore the critical importance of robust security and efficiency in designing authentication protocols for VANETs, paving the way for more secure and scalable vehicular communication systems.

References

[1] Adnan Qayyum, Muhammad Usama, Junaid Qadir, and Ala Al-Fuqaha. Securing connected & autonomous vehicles: Challenges posed by adversarial machine learning and the way forward. IEEE Communications Surveys & Tutorials, 22(2):998–1026, 2020.

[2] Santhosh Kumar Sripathi Venkata Naga, Rajkumar Yesuraj, Selvi Munuswamy, and Kannan Arputharaj. A comprehensive survey on certificateless authentication schemes for vehicular ad hoc networks in intelligent transportation systems. Sensors, 23(5):2682, 2023.

[3] Priyank Sharma, Meet Patel, and Apoorva Prasad. A systematic literature review on internet of vehicles security. arXiv preprint arXiv:2212.08754, 2022.

[4] Harsha Vasudev, Debasis Das, and Athanasios V Vasilakos. Secure message propagation protocols for iovs communication components. Computers & Electrical Engineering, 82:106555, 2020.

[5] Ikram Ali, Alzubair Hassan, and Fagen Li. Authentication and privacy schemes for vehicular ad hoc networks (vanets): A survey. Vehicular Communications, 16:45–61, 2019.

[6] Xiaoxue Liu, Yichuan Wang, Yanping Li, and Hao Cao. Ptap: A novel secure privacy-preserving & traceable authentication protocol in vanets. Computer Networks, 226:109643, 2023.

[7] Yanwei Zhou, Lei Cao, Zirui Qiao, Zhe Xia, Bo Yang, Mingwu Zhang, and Wenzheng Zhang. An efficient identity authentication scheme with dynamic anonymity for vanets. IEEE Internet of Things Journal, 10(11):10052–10065, 2023.

[8] Jyoti Grover. Security of vehicular ad hoc networks using blockchain: A comprehensive review. Vehicular Communications, 34:100458, 2022.

[9] Cong Zhao, Nan Guo, Tianhan Gao, Xinyang Deng, and Jiayu Qi. Pepa: Paillier cryptosystem-based efficient privacy-preserving authentication scheme for vanets.Journal of Systems Architecture, 138:102855, 2023.

[10] Shuyi Chen, Yali Liu, Jianting Ning, and Xiuping Zhu. Basrac: An efficient batch authentication scheme with rule-based access control for vanets. Vehicular Communications, 40:100575, 2023.

[11] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2):120–126, 1978.

[12] Adi Shamir. Identity-based cryptosystems and signature schemes. In Advances in Cryptology: Proceedings of CRYPTO 84 4, pages 47–53. Springer, 1985.

[13] Sattam S Al-Riyami and Kenneth G Paterson. Certificateless public key cryptography. In International conference on the theory and application of cryptology and information security, pages 452–473. Springer, 2003.

[14] Xiaotong Zhou, Min Luo, Pandi Vijayakumar, Cong Peng, and Debiao He. Efficient certificateless conditional privacy-preserving authentication for vanets.IEEE Transactions on Vehicular Technology, 71(7):7863–7875, 2022.

[15] Lunzhi Deng, Bingqin Ning, and Yuhong Jiang. A lightweight certificateless aggregation signature scheme with provably security in the standard model. IEEE Systems Journal, 14(3):4242–4251, 2020.

[16] Huiwen Wang, Liangliang Wang, Kai Zhang, Jinguo Li, and Yiyuan Luo. A conditional privacy-preserving certificateless aggregate signature scheme in the standard model for vanets. IEEE Access, 10:15605–15618, 2022.

[17] Ziyan Gong, Tianhan Gao, and Nan Guo. Pcas: Cryptanalysis and improvement of pairing-free certificateless aggregate signature scheme with conditional privacy-preserving for vanets. Ad Hoc Networks, 144:103134, 2023.

[18] Yangfan Liang and Yining Liu. Analysis and improvement of an efficient certificateless aggregate signature with conditional privacy preservation in vanets. IEEE Systems Journal, 17(1):664–672, 2022.

[19] Wanjun Xiong, Ruomei Wang, Yujue Wang, Yongzhuang Wei, Fan Zhou, and Xiaonan Luo. Improved certificateless aggregate signature scheme against collusion attacks for vanets. IEEE Systems Journal, 17(1):1098–1109, 2022.

[20] Yulei Chen and Jianhua Chen. Cpp-clas: Efficient and conditional privacy-preserving certificateless aggregate signature scheme for vanets. IEEE Internet of Things Journal, 9(12):10354–10365, 2021.

[21] Gowri Thumbur, G Srinivasa Rao, P Vasudeva Reddy, NB Gayathri, DVR Koti Reddy, and M Padmavathamma. Efficient and secure certificateless aggregate signature-based authentication scheme for vehicular ad hoc networks. IEEE Internet of Things Journal, 8(3):1908–1920, 2020.

[22] Eko Fajar Cahyadi and Min-Shiang Hwang. A comprehensive survey on certificateless aggregate signature in vehicular ad hoc networks. IETE Technical Review, 39(6):1265–1276, 2022.

[23] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. Journal of cryptology, 13:361–396, 2000.

Authors' Profiles

Parvin Rastegari was born on June 22, 1986, in Golpayegan, Isfahan, Iran. She received her B.S., M.S., and Ph.D. degrees in electrical engineering from the Department of Electrical and Computer Engineering at Isfahan University of Technology, Isfahan, Iran, in 2008, 2011, and 2019, respectively.

Her M.S. dissertation was in the field of information theory, entitled "The Redundancy of Some Source Codes," and her Ph.D. dissertation focused on cryptography and information security, entitled "Privacy-Preserving Digital Signatures." Since 2020, she has been an assistant professor in the Department of Electrical and Computer Engineering at Golpayegan College of Engineering, Isfahan University of Technology, Golpayegan, Iran. Her current fields of interest include information security, cryptographic protocols, and security challenges in smart grids and the Internet of Things (IoT).

Dr. Rastegari has published several papers in the field of cryptography and information security in various reputable journals and conferences. She is also an active reviewer for several esteemed journals and conferences.

How to cite this paper: Parvin Rastegari, "Authentication in VANETs with Conditional Privacy-Preserving Property Using Certificateless Aggregate Signature Schemes", International Journal of Mathematical Sciences and Computing(IJMSC), Vol.10, No.4, pp. 51-62, 2024. DOI: 10.5815/ijmsc.2024.04.05