Article
A New Conditional Privacy-Preserving Certificateless Aggregate Signature Scheme in the Standard Model for VANETs

Beibei Yuan [1,2], Hui Huang [1] and Chenhuang Wu [2,*]

1 School of Computer Science, Minnan Normal University, Zhangzhou 363000, China; ybb2113@mnnu.edu.cn (B.Y.); hh1162@mnnu.edu.cn (H.H.)
2 Fujian Key Laboratory of Financial Information Processing, Putian University, Putian 351100, China
* Correspondence: wuchenhuang@ptu.edu.cn

Abstract: Vehicular Ad Hoc Networks (VANETs) take moving vehicles and transport facilities as nodes to form mobile networks through wireless communication technology. Its application increases traffic safety and promotes the development of intelligent transport. However, VANETs have security concerns in data transmission. Fortunately, aggregate signature schemes can enhance security and efficiency in the VANETs. Nevertheless, some aggregated signature schemes for VANETs still have security concerns. In this paper, we conduct a security analysis of a conditional privacy-preserving CLAS scheme for VANETs proposed recently. The analysis reveals that the scheme exhibits vulnerabilities to the KGC attack and public key replacement attack. We propose an improved scheme to fix security vulnerabilities in response to these issues. Subsequently, formal and informal security assessments are conducted for the improved scheme, demonstrating that it fulfills security requisites. Furthermore, performance assessment demonstrates the practical viability of the refined scheme.

Keywords: certificateless aggregate signature; conditional privacy-preserving; cryptanalysis; forgery attack; vehicular ad hoc networks

MSC: 94A60

1. Introduction

With the rapid development of network communication technology and automobile industry, the intelligent transport system (ITS) is experiencing a remarkable surge in growth. Meanwhile, vehicle ad hoc networks (VANETs), as an important part of ITS, are also evolving. VANETs are inter-vehicle communication networks with open mobile ad hoc structures. The main composition of VANETs is that vehicles equipped with communication and computing equipment can realize vehicle-to-vehicle (V2V) [1] and vehicle-to-infrastructure (V2I) [2]. All vehicles in the network are equipped with On Board Units (OBUs), which facilitate wireless communication and location function [3]. Vehicles can establish communication with other vehicles and RSUs through OBU [4], which will improve the user's driving experience and safety [5,6]. For example, vehicles can exchange traffic status information in real-time through VANETs, so that drivers can better understand the surrounding traffic conditions and take action in advance against abnormal conditions.

Many challenges remain for vehicular ad hoc networks. Attackers can launch various attacks by intercepting, changing, and forging the location information. For example, malicious vehicles can manipulate traffic information within the network and disseminate false data to create the illusion of road congestion, thus influencing the route choices made by other vehicles. Therefore, it is firstly necessary to ensure the integrity and reliability of received messages to prevent malicious attackers from pretending to be legitimate users to communicate in VANETs. Secondly, the private information of vehicles such as travelling routes and personal identities should also be protected. To address this issue, it can be
solved by anonymous identity. Hubaux et al. [7] proposed the generation of pseudonyms by appropriate authorities. Thus, an anonymous pseudo-identity assigned to the vehicle by the Trace Authority (TRA) can effectively achieve the privacy protection of the vehicle. When a message is disputed, it can ensure that the Traffic Management Centre (TMC) can obtain the real identity of the malicious vehicle and track it to achieve conditional privacy protection of user identity. At the same time, considering the characteristics of high-speed node movement and frequent topology changes in vehicle-mounted self-organizing networks, it is also of great significance to improve the efficiency of each stage of the authentication scheme. Aggregated signatures can achieve the above requirements. An aggregate signature [8] realizes the aggregation of \( n \) different user signatures into an aggregate signature, and the verifier can verify the validity of \( n \) signatures in batches with only one verification, thus effectively reducing the computational cost. The aggregate signature scheme can address the capacity constraints of RSUs and OBUs while achieving message authentication and striking a harmonious equilibrium between security and efficiency.

To solve the privacy of users and security concerns in the VANET environment, researchers have proposed a multitude of certificateless aggregate signature (CLAS) schemes [9–22]. Recently, Wang et al. [23] proposed a CLAS with conditional privacy protection in VANETs. We show that it is vulnerable to KGC attack and public key replacement attack by giving two attacks on Wang et al.’s scheme. In this paper, we propose an improved CLAS scheme to defend against the above security attacks.

Our primary contributions are outlined as follows:

- We analyze a conditional privacy-preserving certificateless aggregate signature scheme in the standard model for VANETs and demonstrate that it is not secure.
- We propose an improved conditional privacy-preserving certificate-free aggregated signature scheme and provide a proof of security.
- The computational overhead and communication overhead of the scheme are simulated in simulation experiments, and the computational overhead and communication of the improved scheme are comparable to the previous CLAS scheme, but more secure than the previous CLAS scheme.

2. Related Work

In 2003, the concept of the aggregate signature was introduced by Boneh et al. [8]. By aggregated signatures, several signatures from a set of messages are consolidated into a single signature, which is equivalent to verifying multiple signatures at once. This not only avoids massive signature transmission storage but also reduces verification overhead. However, identity-based signature schemes suffer from the inherent problem of key escrow. To overcome this obstacle and reduce the burden of certificate management, Al-Riyami and Paterson [24] firstly designed a certificateless encryption scheme in 2003.

In 2007, Castro et al. [25] proposed the first CLAS scheme by combining a certificateless encryption scheme and aggregate signature. But, as the number of signers in the scheme grows, the system overhead will exhibit a linear increase. In that particular year, Gong et al. [9] introduced a pair of CLAS schemes based on bilinear pairings. However, bilinear pairings are computationally expensive, making them unsuitable for resource-constrained environments. Subsequently, Xiong et al. [10] developed a CLAS scheme incorporating a immutable pairing operation to reduce the computational burden, and established its security under the random oracle model. However, some scholars [11,12,26] demonstrated that the scheme of Xiong et al. [10] is incapable of resisting a type II adversary attack, anti-collision attack and internal attack.

Malhi and Batra [14] introduced an aggregate signature scheme based on certificateless VANETs in 2015, characterized by constant pairing computations. Afterward, Kumar and Sharma [15] indicated that it is vulnerable to a type II attacker, and enhanced a safer CLAS scheme. In 2019, Zhong et al. [27] constructed a new CLAS authentication protocol by combining a full aggregation in VANETs. Kamil and Ogundoyin [28] showed that the scheme was incapable of defending against type II attacks. So, they designed a safer
and enhanced CLAS scheme to deal with these attacks. In 2020, to enhance data sharing efficiency within VANET systems, Cui et al. [29] introduced a data download scheme for privacy-preserving VANETs based on edge computing, which provides a security proof under the random oracle model. In the same year, Xu et al. [22] proposed a new CLAS scheme to solve the problem of routing security authentication. In the next year, Kamil et al. [30] introduced a group key agreement to make it more efficient in the IoV. The group key distribution mechanism facilitates efficient group communication while accommodating dynamic updates. In 2022, Cao et al. [31] proposed lattice-based group signatures that are resistant to quantum attacks. Zhang et al. [32] proposed a certificateless signature based on a homomorphic hash function, which is applied in an auditing scheme to achieve conditional privacy protection this year. In 2023, Gong et al. [33] proposed a pairing-free PCAS scheme without bilinear pair operations to make the scheme more secure and efficient. This year, Xu et al. [34] proposed the PAASH+ scheme that can resist public key substitution attacks to achieve privacy protection in medical scenarios. Li et al. [35] also designed a CPPA scheme by introducing linkable group signatures. The scheme protects privacy and provides authentication, which improves the trustworthiness and traceability of messages. More recently, Wang et al. [23] proposed a CLAS scheme for VANETs within conditional privacy-preservation. However, Shim et al. [36] attacked the scheme and proved that it is not safe against KGC attacks, suffering from logical errors. We indicate that this scheme not only suffers from the above security problems but also fails to resist the public key replacement attack in this paper. Meanwhile, we propose a new improved scheme to resist these attacks.

3. Review of Wang et al.'s CLAS Scheme

In this section, we provide a concise overview of the CLAS scheme proposed by Wang et al. [23].

3.1. System Infrastructure

The CLAS scheme consists of eight phases and five entities in the CLAS scheme, which include Key Generation Center (KGC), Trace Authority (TRA), On board Units (OBUs), Roadside Units (RSUs), and Traffic Management Center (TMC). As shown in Figure 1, we will provide a description of the following five entities.

Key Generation Center (KGC): KGC collaborates with TRA to generate public parameters for VANET to ensure strong security. In addition, KGC generates partial private keys for vehicles.

Trace Authority (TRA): TRA performs key tasks of setup algorithms and vehicle registration within VANETs. As part of this process, TRA allocates a pseudo-identity to each vehicle upon its entry into the network. It is important that only TRA possesses knowledge of the true vehicle identity to ensure safety. In the event of an occurrence of malicious traffic behavior by a specific vehicle, TRA has the capacity to reveal the authentic identity of the mentioned malevolent vehicle.

On Board Units (OBUs): Each vehicle on the road has an On Board Unit (OBU) that allows communication via V2V interactions and V2I communications with Roadside Units (RSUs). Individual pseudo-markers are used to transmit traffic-related data and signatures from vehicles to adjacent RSUs.

Roadside Units (RSUs): RSUs use a DSRC protocol for V2I communication within their coverage areas along roadways. Specifically, RSUs undertake the task of validating individual traffic-related messages emanating from OBUs. After the RSU establishes the legitimacy of the traffic-related message from an OBU, it generates an aggregate signature and transmits it to the TMC.

Traffic Management Center (TMC): TMC decides whether to accept or reject the aggregated signature and extracts insights on the current traffic conditions. Therefore, TMC plays a crucial role in regulating and managing traffic flow.
3.2. Threat Model

In the realm of VANETs, two distinct categories of attackers emerge, external attackers $A_1$ and internal attackers $A_2$. Attacker $A_1$ has the ability to request the user’s public key or substitute it, and remains unaware of the system master key. Attacker $A_2$ can obtain the system master key but is unable to alter the public key or query the public key. The former operates externally to the VANETs’ ecosystem, while the latter comprises entities within the VANETs’ network. Given the vulnerability of public wireless networks, all adversaries possess the capability to intercept vehicular-RSU communications, enabling them to engage in eavesdropping, interception, modification, or deletion of transmitted information. Notably, our assumptions hold the TMC, KGC, and TRA to be entities with full credibility. Vehicles and RSUs are honest but curious agents and semi-trusted entities, respectively. This implies that they strictly adhere to predetermined protocols while being curious about extracting privacy-related attributes (such as identity, velocity, and location) from accessible data. It is worth noting that any adversary cannot obtain the vehicle’s key. Lastly, the temporal synchronization across all VANETs components is maintained.

3.3. Wang et al.’s CLAS Scheme

The eight stages are described as follows. In addition, Table 1 shows some useful symbols from the CLAS scheme of Wang et al. [23].

- **Setup:** TRA and KGC select a bilinear map $e : G_1 \times G_1 \rightarrow G_2$ with prime order $q > 2^v$, where $v$ is a security parameter. KGC randomly chooses $P, Q \in G_1$, $s \in Z_q^*$ and calculates $P_{\text{pub}} = sP$. TRA randomly chooses $k$ and calculates $K = kP$. The secret key $s$ and $k$ are kept secretly. Then, TRA and KGC choose three hash
functions $H_1 : G_1 \rightarrow Z_q^*$, $H_2 : \{0, 1\}^* \times G_1 \rightarrow Z_q^*$, $H_3 : \{0, 1\}^* \times \{0, 1\}^* \times G_1^6 \rightarrow Z_q^*$. Finally, the public parameters of the publishing system are: $\text{params} = \{G_1, G_2, q, e, P, Q, p_{pub}, H_1, H_2, H_3\}$.

Table 1. Notations in Wang et al.’s CLAS scheme.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q$</td>
<td>A secure prime number</td>
</tr>
<tr>
<td>$v$</td>
<td>Security parameter</td>
</tr>
<tr>
<td>$G_1$</td>
<td>An additive cyclic group</td>
</tr>
<tr>
<td>$G_2$</td>
<td>A multiplicative cyclic group</td>
</tr>
<tr>
<td>$e$</td>
<td>A bilinear map</td>
</tr>
<tr>
<td>$P, Q$</td>
<td>Two generators of the group $G_1$</td>
</tr>
<tr>
<td>$\text{params}$</td>
<td>System public parameters</td>
</tr>
<tr>
<td>$p_{pub}$</td>
<td>The public key of system</td>
</tr>
<tr>
<td>$s$</td>
<td>The master secret key of system</td>
</tr>
<tr>
<td>$H_{t_1}, H_{t_2}, H_3$</td>
<td>One way hash functions</td>
</tr>
<tr>
<td>$k$</td>
<td>Identity tracking key</td>
</tr>
<tr>
<td>$V_{t_1}$</td>
<td>the $i^{th}$ new vehicle</td>
</tr>
<tr>
<td>$ID_i$</td>
<td>The real identity of the vehicle $V_{t_1}$</td>
</tr>
<tr>
<td>$PID_i$</td>
<td>A set of pseudonyms of $V_{t_1}$</td>
</tr>
<tr>
<td>$PID_{i,j}$</td>
<td>The $j^{th}$ pseudonym of $V_{t_1}$</td>
</tr>
<tr>
<td>$d_i$</td>
<td>Partial private Key of $V_{t_1}$</td>
</tr>
<tr>
<td>$x_i$</td>
<td>Secret value of $V_{t_1}$</td>
</tr>
<tr>
<td>$PK_i = (X_i, R_i)$</td>
<td>Public key of $V_{t_1}$</td>
</tr>
<tr>
<td>$SK_i = (d_i, x_i)$</td>
<td>Private key of $V_{t_1}$</td>
</tr>
<tr>
<td>$m_i$</td>
<td>Traffic-related message</td>
</tr>
<tr>
<td>$TS_i$</td>
<td>Current timestamp chosen by $V_{t_1}$</td>
</tr>
<tr>
<td>$\sigma_i = (U_i, V_i, W_i)$</td>
<td>Signature on a message $m_i$</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>An aggregate signature</td>
</tr>
</tbody>
</table>

- **Pseudonym Generation**: First, the TRA will designate a pseudonym $PID_{i,j}$ to the new vehicle $V_{t_1}$. Vehicle $V_{t_1}$ randomly chooses $t_{i,j} \in Z_q^*$ and calculates $T_{i,j} = t_{i,j}P$. Then, vehicle $V_{t_1}$ sends $(ID_{i,j}, T_{i,j})$ to the TRA in secret. The TRA verifies the validity of the $ID_{i,j}$ and calculates $PID_{i,1,j} = ID_{i,j} \oplus H_1(kP + T_{i,j})$ and $PID_{i,j} = (PID_{i,1,j}, T_{i,j})$. Afterwards, TRA transmits $PID_{i,j}$ to vehicle $V_{t_1}$. Through obtaining the pseudonym $PID_{i,j} = (PID_{i,1,j}, T_{i,j})$ to calculate $ID_{i,j} = PID_{i,1,j} \oplus H_1(kP + T_{i,j})$, TRA can effectively determine the true identity of the vehicle when the vehicle $V_{t_1}$ is involved in a malicious collision.

- **Partial Private Key Generation**: First, KGC randomly chooses $r_i \in Z_q^*$ and calculates $R_i = r_iP$. It also calculates $k_i = H_2(PID_{i,j}, R_i)$ and $d_i = r_i + k_i \times q$. $d_i$ is the partial private key for the vehicle $V_{t_1}$. Subsequently, KGC securely transmits the partial private key $d_i$ to vehicle $V_{t_1}$ via a trusted message route.

- **Public/Private Key Generation**: After receiving a message from the KGC, the vehicle $V_{t_1}$ chooses a single secret value $x_i \in Z_q^*$. The vehicle calculates $X_i = x_iP$ and public key of the vehicle represented as $PK_i = (X_i, R_i)$. Furthermore, $(d_i, x_i)$ is denoted as the value of the private key.

- **Signature Generation**: Firstly, the OBU selects the present timestamp $TS_i$. The OBU randomly selects $u_i \in Z_q^*$ and calculates $U_i = u_iP$ and $V_i = u_iQ$. Following this, the vehicle $V_{t_1}$ calculates $h_i = H_3(m_i||TS_i, PID_{i,j}, U_i, V_i, W_i, PK_i)$ and $W_i = (d_i + h_i x_i)Q + V_i$. The signature $\sigma_i = (U_i, V_i, W_i)$ is generated on $m_i||TS_i$ and $(m_i, TS_i, PK_i, PID_{i,j}, U_i, V_i, W_i)$ is communicated to the RSU. Whenever a vehicle $V_{t_1}$ transmits a signature, TRA generates a new pseudonym $PID_{i,j}$ and assigns it to $V_{t_1}$.

- **Single Signature Verification**: Upon receipt of the signature $\sigma_i$ on $m_i||TS_i$, the respective RSU involves firstly assessing the timeliness of the timestamp $TS_i$. If $TS_i$ is valid, the RSU proceeds to validate the signature’s authenticity, as detailed below. The RSU
computes $k_i = H_2(PID_{ij}, R_i)$ and $h_i = H_3(m_i||TS_{ij}, PID_{ij}, U_i, V_i, W_i, PK_i)$ and verifies whether (1) is established.

$$e(W_i, P) = e(R_i + k_iP_{pub} + h_iX_i + U_i, Q)$$  \(1\)

Upon (1) holds, the single signature $\sigma_i$ on $m_i||TS_i$ is accepted by the RSU; conversely, it results in rejection.

- **Aggregate**: Upon the receipt of a set of $n$ distinct signatures $\sigma_i$ pertaining to diverse messages $m_i||TS_i$ from distinct vehicles $Vh_i$, the RSU calculates $U = \sum_{i=1}^{n} U_i$, $V = \sum_{i=1}^{n} V_i$, and $W = \sum_{i=1}^{n} W_i$. Subsequently, the RSU transmits the aggregate signature $\sigma = (U, V, W)$ to the TMC.

- **Aggregate Verification**: After the reception of the aggregated signature $\sigma$ and corresponding tuples $(m_i, TS_i, PID_{ij}, PK_i)$, TMC examines the temporal freshness of each timestamp $TS_i(i = 1, 2, \ldots, n)$ initially. Subsequent to verification, the TMC computes $k_i = H_2(PID_{ij}, R_i)$ and $h_i = H_3(m_i||TS_i, PID_{ij}, U_i, V_i, W_i, PK_i)$. Lastly, the TMC verifies whether (2) is established.

$$e(W_i, P) = e(\sum_{i=1}^{n} R_i + \sum_{i=1}^{n} k_iP_{pub} + \sum_{i=1}^{n} h_iX_i + U_i, Q)$$  \(2\)

If (2) holds, the aggregate signature $\sigma_i$ on $m_i||TS_i(i = 1, 2, \ldots, n)$ is accepted by the TMC; conversely, it results in rejection.

4. Cryptanalysis of Wang et al.’s CLAS Scheme

We demonstrate the presence of several kinds of attack in Wang et al.’s CLAS scheme [23].

4.1. Incorrectness of the Signature Generation

In the Signature Generation algorithm, the vehicle $Vh_i$ calculates $h_i = H_3(m_i||TS_i, PID_{ij}, U_i, V_i, W_i, PK_i)$ and $W_i = (d_i + h_iX_i)Q + V_i$ to generate the single signature. However, $h_i$ must be calculated before vehicle $Vh_i$ calculates $W_i$, which contradicts the use of $W_i$ in the computation of $h_i$. Therefore, the Signature Generation algorithm is logically incorrect.

To resolve the issue in the Signature Generation algorithm, let the vehicle $Vh_i$ calculate $h_i = H_3(m_i||TS_i, PID_{ij}, U_i, V_i, W_i, PK_i)$ firstly, and then calculate $W_i = (d_i + h_iX_i)Q + V_i$.

4.2. KGC Forge Attack

In KGC forge attacks, we know that $Q$ and $P$ are chosen by the KGC in the Setup algorithm. Thus, KGC knows the discrete logarithm of $Q$ relative to $P$, assuming this discrete logarithm is $l$. We show that KGC has the ability to generate a forged signature for any message from the RSU, which can be verified.

- KGC randomly selects $u'_i \in \mathbb{Z}_q^*$ and calculates $U'_i = u'_iP$ and $V'_i = u'_iQ$. KGC picks any message $m'_i$.
- KGC computes $k_i = H_2(PID_{ij}, R_i)$, $h'_i = H_3(m'_i||TS_i, PID_{ij}, U'_i, V'_i, W'_i, PK_i)$ and $W'_i = l(R_i + k_iP_{pub} + h'_iX_i + U'_i)$.
- KGC outputs $\sigma'_i = (U'_i, V'_i, W'_i)$.

It is easy to prove that $\sigma'_i = (U'_i, V'_i, W'_i)$ can be verified by the RSU using the Single Signature Verification algorithm. Here, the validity verification process of the signature is as follows.

$$e(W'_i, P) = e(l(R_i + k_iP_{pub} + h'_iX_i + U'_i), P)$$

$$= e(R_i + k_iP_{pub} + h'_iX_i + U'_i, lP)$$

$$= e(R_i + k_iP_{pub} + h'_iX_i + U'_i, Q)$$

Therefore, the forged signature $\sigma'_i$ passes the Single Signature Verification algorithm.
4.3. Replace Public Key Attack

We show the vulnerability of Wang et al.’s scheme against the public key replacement attack. Specifically, we highlight that an adversary can generate legitimate signatures for arbitrary messages pertaining to any vehicles using solely a single authentication message. The details are as follows.

- Computes $k_i = H_2(PID_{i,j}, R_i)$.
- The adversary chooses a secret value $x'_i \in Z_q^*$ and calculates $X'_i = x'_iP$ to replace the public key $X_i$. The public key of vehicle $V_{h_i}$ is replaced as $PK_i = (X'_i, R_i)$.
- The adversary picks a message, $m'_i$. The adversary randomly selects $u'_i \in Z_q^*$ and constructs $U'_i = u'_iP - (R_i + k_iP_{pub})$. Then, the adversary calculates $h'_i = H_3(m'_i || TS_j, PID_{i,j}, U'_i, V'_i, PK_i)$ and constructs $W'_i = (h'_iX'_i + u'_i)Q$. Then, it outputs $\sigma'_i = (U'_i, W'_i)$.

We note that $e(W_i, P) = e(R_i + k_iP_{pub} + h_iX_i + U_i, Q)$ in the verification. The forgery process is as follows.

$$
e(R_i + k_iP_{pub} + h'_iX'_i + U'_i, Q) = e(R_i + k_iP_{pub} + h'_iX'_i + (u'_iP - (R_i + k_iP_{pub})), Q)$$
$$= e(h'_iX'_i + u'_iP, Q)$$
$$= e((h'_iX'_i + u'_i), Q, P)$$
$$= e(W'_i, P)$$

Therefore, the adversary replaces the public key and forges a signature $\sigma'_i$ on message $m'_i$ that can pass the Single Signature Verification algorithm.

5. Improvement for Wang et al.’s CLAS Scheme

The improved CLAS scheme includes eight distinct stages. Additionally, Table 2 presents partial essential notations within the improved CLAS scheme, and others are listed in Table 1.

Table 2. Notations in improved CLAS scheme.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_{pub}$</td>
<td>The public key of system</td>
</tr>
<tr>
<td>$H_1, H_2, H_3, H_4, H_5$</td>
<td>One-way hash functions</td>
</tr>
<tr>
<td>$Z$</td>
<td>Hash value of the system public key</td>
</tr>
<tr>
<td>$\sigma_i = (U_i, W_i)$</td>
<td>Signature on a message $m_i$</td>
</tr>
<tr>
<td>$\sigma = (U, W)$</td>
<td>An aggregate signature</td>
</tr>
</tbody>
</table>

- **Setup:** TRA and KGC generate a prime order $q > 2^v$ by entering the safety parameter $v$. Subsequently, the additive cyclic group $G_1$ and multiplicative cyclic groups $G_2$ are generated with prime order $q > 2^v$. A bilinear map $e : G_1 \times G_1 \rightarrow G_2$ is selected. TRA and KGC choose five hash functions $H_1 : G_1 \rightarrow Z_q^*, H_2 : \{0,1\}^* \times G_1 \times G_1 \rightarrow Z_q^*$, $H_3 : G_1 \rightarrow G_1$, $H_4 : G_1^* \rightarrow Z_q^*$, $H_5 : \{0,1\}^* \times \{0,1\}^* \times G_1^* \rightarrow Z_q^*$. Then, KGC randomly chooses $P \in G_1$ and $s \in Z_q^*$ and calculates $y_{pub} = sP$ and $Z = H_3(y_{pub})$. TRA randomly chooses $k$ and calculates $K = kP$. Finally, the public parameters of the publishing system are $\text{params} = \{G_1, G_2, q, e, P, Z, y_{pub}, H_1, H_2, H_3, H_4, H_5\}$, where the master secret key $s$ and the identity tracking key $k$ are kept secretly.

- **Pseudonym Generation:** The vehicle is required to undergo registration with the TRA to ensure the security of the user’s information before it can transmit information in VANETs. The virtual identity $ID_i$ of the vehicle achieves anonymity during communication. The TRA will designate a pseudonym as $PID_{i,j}$, which represents the $j$-th pseudonymous identifier assigned to the vehicle $V_{h_i}$. Vehicle $V_{h_i}$ randomly chooses $t_{i,j} \in Z_q^*$ and calculates $T_{i,j} = t_{i,j}P$. Then, vehicle $V_{h_i}$ sends $(ID_i, T_{i,j})$ to the TRA in secret. The TRA verifies the validity of the $ID_i$ and calculates $PID_{i,1,j} = ID_i \oplus H_1(kP + T_{i,j})$ and $PID_{i,j} = \langle PID_{i,1,j}, T_{i,j} \rangle$. Afterwards, TRA transmits
$PID_{ij}$ to vehicle $Vh_i$. In the event that vehicle $Vh_i$ is involved in malevolent collision, the TRA possesses the capability to trace its actual identity $ID_i$. After obtaining the pseudonym $PID_{ij} = (PID_{i,j}, T_{ij})$ to calculate $ID_i = PID_{i,j} \oplus H_1(kP + T_{ij})$, TRA can effectively determine the true identity of the vehicle when the vehicle $Vh_i$ is involved in a malicious collision.

- **Partial Private Key Generation**: By obtaining the $params$ along with the master key $s$, KGC generates the partial private key $d_i$ for the vehicle $Vh_i$, using the following process. KGC randomly selects $r_j \in Z_q^*$ and calculates $R_i = r_jP$. Additionally, $k_i = H_2(PID_{ij}, R_i)$ and $d_i = r_i + k_i s \mod q$ are derived. Note that $d_i$ is assigned as the partial private key for vehicle $Vh_i$. Following this computation, KGC securely transmits the partial private key $d_i$ to vehicle $Vh_i$ via a trusted message route.

- **Public/Private Key Generation**: After receiving a message from the KGC, the vehicle $Vh_i$ chooses a single secret value $x_i \in Z_q^*$. Specifically, the vehicle calculates $X_i = x_iP$, and the public key of the vehicle is represented as $PK_i = (X_i, R_i)$. Furthermore, $(d_i, x_i)$ is denoted as the value of the private key.

- **Signature Generation**: The process of generating a signature for a traffic-related message $m_i \in Z_q^*$ is as follows.

(1) The OBU selects the present timestamp $TS_i$.

(2) The OBU randomly chooses $u_i \in Z_q^*$ and calculates $U_i = u_iP$.

(3) The vehicle $Vh_i$ calculates $\phi_i = H_4(y_{pub}, PK_i, U_i)$, $h_i = H_5(m_i || TS_i, PID_{ij}, U_i, PK_i)$ and $W_i = (d_i, \phi_i + h_i x_i + u_i)Z$.

The signature $\sigma_i = (U_i, W_i)$ is generated on $m_i || TS_i$ and $(m_i, TS_i, PK_i, PID_{ij}, U_i, W_i)$ is sent to the RSU. Whenever a vehicle $Vh_i$ transmits a signature, a TRA generates a new pseudonym $PID_{ij}$ and assigns it to $Vh_i$. This ensures a single use of each pseudonym, and the vehicle $Vh_i$ substitutes the former pseudonym with the updated one.

- **Single Signature Verification**: Upon the signature $\sigma_i$ on $m_i || TS_i$, the respective RSU firstly involves assessing the timeliness of the timestamp $TS_i$. If $TS_i$ is on validity, the RSU proceeds to validate the signature’s authenticity, as detailed below. The RSU calculates $k_i = H_2(PID_{ij}, R_i)$, $\phi_i = H_4(y_{pub}, PK_i, U_i)$ and $h_i = H_5(m_i || TS_i, PID_{ij}, U_i, PK_i)$ and verifies whether (3) is established.

$$e(W_i, P) = e((R_i + k_i y_{pub}) \phi_i + h_i X_i + U_i, Z) \quad (3)$$

If (3) holds, the singular signature $\sigma_i$ on $m_i || TS_i$ is accepted by the RSU; conversely, it results in rejection.

- **Aggregate**: When receiving a set of $n$ distinct signatures, $\sigma_i$ of diverse messages $m_i || TS_i$ from distinct vehicles $Vh_i$. The RSU calculates $U = \sum_{i=1}^{n} U_i$ and $W = \sum_{i=1}^{n} W_i$. Afterward, the RSU transmits the aggregate signature $\sigma = (U, W)$ to the TMC.

- **Aggregate Verification**: After the reception of the aggregated signature $\sigma$ and corresponding tuples $(m_i, TS_i, PID_{ij}, PK_i)$, TMC examines the temporal freshness of each timestamp $TS_i(i = 1, 2, \ldots, n)$. Next, the TMC computes $k_i = H_2(PID_{ij}, R_i)$, $\phi_i = H_4(y_{pub}, PK_i, U_i)$ and $h_i = H_5(m_i || TS_i, PID_{ij}, U_i, PK_i)$. Lastly, the TMC verifies whether (4) is established.

$$e(W, P) = e(\sum_{i=1}^{n} (R_i + k_i y_{pub}) \phi_i + \sum_{i=1}^{n} h_i X_i + U, Z) \quad (4)$$

If (4) holds, the aggregate signature $\sigma_i$ on $m_i || TS_i(i = 1, 2, \ldots, n)$ is accepted by the TMC; conversely, it results in rejection.

**Remark 1.** When there are a few corrupted signatures in the aggregated signature, it is necessary to go through them one by one to verify and lock the invalid signatures. In order to improve the efficiency of retrieving the corrupted signatures, the bisection method can be used to lock the invalid signatures quickly. Meanwhile, for the vehicle that often generates invalid signatures, a penalty
mechanism can be set to delay the verification of the vehicle or verify it individually. In turn, the effectiveness and efficiency of aggregated signatures in batch verification is improved.

**Remark 2.** Pseudonym Generation, Partial Private Key Generation and Public/Private Key Generation algorithms can be predefined in advance.

6. Security Analysis

Firstly, the correctness proof of **Single Signature Verification** and **Aggregate Verification** is explained in this section. Meanwhile, we conduct a formal and informal security analysis of the improved CLAS scheme. Finally, we indicate its capability to fulfill security requirements within VANETs.

6.1. Correctness

The correctness of the **Single Signature Verification** algorithm is described below.

\[
e(W_i, P) = e((d_i\phi_i + h_i x_i + u_i)Z, P) = e(((r_i + k_i s_\phi_i + h_i x_i + u_i)Z, P) = e((R_i + k_i y_{pub})\phi_i + h_i X_i + U_i, Z)
\]

The correctness of the **Aggregate Signature Verification** algorithm is described below.

\[
e(W, P) = e(\sum_{i=1}^{n} W_i, P) = e(\sum_{i=1}^{n} ((d_i\phi_i + h_i x_i + u_i)Z, P) = e(\sum_{i=1}^{n} d_i\phi_i + \sum_{i=1}^{n} h_i x_i + \sum_{i=1}^{n} u_i)Z, P) = e(\sum_{i=1}^{n} (R_i + k_i y_{pub})\phi_i + \sum_{i=1}^{n} h_i X_i + \sum_{i=1}^{n} U_i, Z)
\]

6.2. Formal Security Analysis

The formal security proof of the improved scheme in a standard model is provided in this section. We consider two kinds of attackers, \(A_1\) and \(A_2\): an external attacker \(A_1\) can substitute the vehicle’s public key, and is not capable of corroding the KGC’s system master key; an internal attacker \(A_2\) can corrode the KGC’s system master key, but is incapable of substituting the vehicle’s public key.

**Theorem 1.** In the standard model, the proposed CLAS scheme is unforgeable when the CDHP assumption holds in the adaptive chosen-identity attacks (EUF-CMA) against Adversary \(A_1\).

**Lemma 1.** In the CLAS scheme, challenger \(C\) can solve the Computational Diffie–Hellman Problem (CDHP) if the adversary \(A_1\) succeeds in producing valid forged signatures in game I in the standard model.

**Proof.** Suppose a random tuple \((P, aP, bP)\) representing the Computational Diffie–Hellman Problem (CDHP) is given. Let \(PID_i\) be the challenge identity. If \(A_1\) produces a valid signa-
ture in the improved CLAS scheme, subsequent to their interaction with $A_1$, the challenger $C$ acquires the value of $abP$.

**Setup:** Challenger $C$ executes the Setup algorithm to generate system public parameters by a security parameter $v$ with sets $Z = bP$, and publishes system public parameters $\text{params} = \{G_1, G_2, q, e, P, Z, y_{pub}, H_1, H_2, H_3, H_4, H_5\}$. Then, $C$ sends these system parameters to $A_1$, and the master secret key $s$ is kept secretly.

**Queries:** $A_1$ executes the following queries and interacts with challenger $C$. Challenger $C$ maintains lists $L_U$ and $L_P$, which are initially empty. $A_1$ performs user public key queries, which takes precedence over other queries.

- **User public key queries:** Challenger $C$ maintains the list $L_U = (PID_{ij}, r_i, x_i)$. Given a request with pseudonym $PID_{ij}$, challenger $C$ will search $(PID_{ij}, r_i, x_i)$ in $L_U$. If successful, $C$ returns $(r_iP, x_iP)$. Instead, $C$ discusses the following situations.
  
  1. If $PID_{ij} = PID_T$, $C$ randomly selects $x_i$ and assigns $R_i = aP$. Subsequently, $(PID_{ij}, r_i, x_i)$ is appended to the list $L_U$, where $\bot$ represents a null value. Following this, $C$ transmits $PK_i = (R_i, x_iP)$ to $A_1$.
  2. If $PID_{ij} \neq PID_T$, $C$ randomly selects $x_i$, $r_i$ and assigns $X_i = x_iP, R_i = r_iP$. Subsequently, $(PID_{ij}, r_i, x_i)$ is appended to the list $L_U$. Following this, $C$ transmits $PK_i = (R_i, x_iP)$ to $A_1$.

- **User public key replacement queries:** Challenger $C$ holds list $L_R = (PID_{ij}, PK_i, PK'_i)$; when $A_1$ requests to query the tuple $(PID_{ij}, PK_i)$, $C$ substitutes $PK_i$ with $PK'_i$ and adds $(PID_{ij}, PK_i, PK'_i)$ to $L_P$.

- **Partial private key extraction queries:** Upon $A_1$’s submission of a request using the pseudonym $PID_{ij}$, challenger $C$ conducts a search within $L_P$ for $(PID_{ij}, d_i)$. If the search is successful within $L_P$, $C$ will return $d_i$ to $A_1$. In the case of failure, $C$ proceeds with the instructions, as follows.
  
  1. If $PID_{ij} = PID_T, C$ fails and ends.
  2. If $PID_{ij} \neq PID_T, C$ searches for $r_i$ in the list $L_U$ and calculates $d_i = r_i + k_is \mod q$. Then, $C$ transmits $d_i$ to $A_1$.

- **Secret value queries:** $A_1$ requests with the pseudonym $PID_{ij}$, challenger $C$ searches $x_i$ in $L_U$, and returns $x_i$ to $A_1$.

- **Signature queries:** After receiving the query for the tuple $(PID_{ij}, m_i || TS_i)$ from $A_1$, $C$ performs user public key queries, partial private key extraction queries and secret value queries to obtain the values of $R_i, d_i, x_i$. After that, $C$ computes $t_i = H_2(PID_{ij}, R_i), \phi_i = H_4(y_{pub}, PK_i, U_i)$ and $h_i = H_5(m_i || TS_i, PID_{ij}, U_i, PK_i)$. $C$ randomly selects $u_i$ then calculates $U_i = U_iP$ and $W_i = (d_i\phi_i + h_i|x_i + u_i)Z$. Finally, $C$ outputs $s_i = (U_i, W_i)$ to $A_1$ as the signature on the tuple of $(PID_{ij}, m_i || TS_i)$; such a signature is valid.

**Forgery Phase:** $A_1$ forges an aggregate signature $s'_i = (U'_i, W'_i)$ on message $m'_i || TS'_i$ and outputs it. After $C$ obtains the forged signature $s'_i$, if $PID_{ij} \neq PID_T$, the game aborts. Otherwise, $PID_{ij} = PID_T$, and there are $PK'_i = (aP, x'_iP)$ and $Z = bP$. $C$ looks for the list $L_U$ to obtain $x'_i$, and calculates $k'_i = H_2(PID_T, aP), \phi'_i = H_4(y_{pub}, PK'_i, U'_i)$ and $h'_i = H_5(m'_i || TS'_i, PID_T, U'_i, PK'_i)$. Due to $s'_i$ being a valid signature, $U'_i = U_iP$ and $W'_i = (d'_i\phi'_i + h'_i|x'_i + u'_i)Z = ((a + k'_is)\phi'_i + h'_i|x'_i + u'_i)bP$. Therefore, $C$ calculates $abP = \phi^{-1}_{i}(W'_i - (u'_i + h'_ix'_i)bP) - k'_sbP$ as the solution of CDHP.

Likewise, $A_1$ outputs a forged aggregate signature $s'' = (U', W')$ on the message $m''_i || TS''_i (i = 1, 2, \ldots, n)$, where $U'' = \sum_{i=1}^{n} U'_i$ and $W'' = \sum_{i=1}^{n} W'_i$, $s''$ is the forged signature of user $PID_{ij}$ on $m''_i || TS''_i$, who has not been executed for Partial private key extraction queries. If $PID_{ij} = PID_T, PK'_i = (aP, x'_iP)$, and $Z = bP$. Subsequently, $C$ performs the following process to solve CDHP.

- Compute $\phi'_i = H_4(y_{pub}, PK'_i, U'_i)$ for $(i = 1, 2, \ldots, n)$.
- Look for $r'_i$ in the list $L_{U_i}$ and calculate $k'_i = H_2(PID_{ij}, r'_iP)$ and $d'_i = r'_i + k'_is \mod q$ for $i \neq \tau$. 
In the standard model, the proposed CLAS scheme is unforgeable when the CDHP problem is given. If \( A_2 \) produces a signature that passes verification within the improved CLAS scheme, subsequent to their interaction with \( A_2 \), the challenger \( C \) acquires the value of \( abP \).

**Proof.** Suppose a random tuple \((P, aP, bP)\) representing the Computational Diffie–Hellman Problem (CDHP) is given. If \( A_2 \) produces a signature that passes verification within the improved CLAS scheme, subsequent to their interaction with \( A_2 \), the challenger \( C \) acquires the value of \( abP \).

**Setup:** Challenger \( C \) executes the Setup algorithm to generate system public parameters by a security parameter \( \nu \) and sets \( Z = bP \), and publishes system public parameters \( \text{params} = \{G_1, G_2, \eta, e, P, Z, y_{pub}, H_1, H_2, H_3, H_4, H_5\} \). Then, \( C \) sends these system parameters and the master secret key \( s \) to \( A_2 \).

**Queries:** \( A_2 \) executes the following queries and interacts with challenger \( C \). Firstly, challenger \( C \) maintains the empty list \( L_U \). Then, \( A_2 \) performs user public key queries, which takes precedence over other queries.

- User public key queries: Challenger \( C \) keeps the list \( L_U \), where \( L_U = (\text{PID}_{ij}, r_i, x_i) \). When presented with a request with the pseudonym \( \text{PID}_{ij} \), \( C \) conducts a search within \( L_U \) for \( (\text{PID}_{ij}, r_i, x_i) \). Upon a successful match, \( C \) returns \( (r_iP, x_iP) \). Alternatively, \( C \) analyzes the following two situations.

  1. If \( \text{PID}_{ij} = \text{PID}_r \), \( C \) randomly selects \( r_i \) and assigns \( X_i = aP \). Subsequently, \( (\text{PID}_{ij}, r_i, \bot) \) is appended to the list \( L_U \). Following this, \( C \) transmits \( PK_i = (r_iP, X_i) \) to \( A_2 \).

  2. If \( \text{PID}_{ij} \neq \text{PID}_r \), \( C \) randomly selects \( x_i, r_i \) and assigns \( X_i = x_iP, R_i = r_iP \). Subsequently, \( (\text{PID}_{ij}, r_i, x_i) \) is appended to the list \( L_U \). Following this, \( C \) transmits \( PK_i = (R_i, X_i) \) to \( A_2 \).

- Secret value queries: \( A_2 \) submits a query for the pseudonym \( \text{PID}_{ij} \). If \( \text{PID}_{ij} = \text{PID}_r \), challenger \( C \) fails and aborts. Moreover, \( C \) seeks \( x_i \) in the list \( L_U \) and returns \( x_i \).

- Signature queries: After \( A_2 \) requests the query of a tuple \( (\text{PID}_{ij}, m_i || TS_i) \), \( C \) performs User public key queries, Partial private key extraction queries and Secret value queries to obtain the values of \( R_i, d_i, x_i \). Afterwards, \( C \) calculates \( k_i = H_2(\text{PID}_{ij}, R_i) \), \( \phi_i = H_4(y_{pub}, PK_i, U_i) \), \( d_i = r_i + k_i s \mod q \) and \( h_i = H_5(m_i || TS_i, \text{PID}_{ij}, U_i, PK_i) \). \( C \) randomly selects \( u_i \) and calculates \( U_i = u_iP \) and \( W_i = (d_i \phi_i + h_i x_i + u_i)Z \). At last, \( C \) outputs \( \sigma_i = (U_i, W_i) \) to \( A_2 \), and as the signature on the last \( (\text{PID}_{ij}, m_i || TS_i) \), such a signature is valid.

**Forgery Phase:** \( A_2 \) forges an aggregate signature \( \sigma'_i = (U'_i, W'_i) \) on message \( m'_i || TS'_i \) and outputs it. After \( C \) obtains the forged signature \( \sigma'_i \), if \( \text{PID}_{ij} \neq \text{PID}_r \), the game aborts. Otherwise, \( \text{PID}_{ij} = \text{PID}_r \), so there are \( PK'_i = (r'_iP, aP) \) and \( Z = bP \). \( C \) looks for the list \( L_U \) to obtain \( r'_i \) and calculates \( k'_i = H_2(\text{PID}_r, R'_i) \), \( \phi'_i = H_4(y_{pub}, PK_i, U'_i) \) and \( h'_i = H_5(m'_i || TS'_i, \text{PID}_r, U'_i, PK'_i) \). \( \sigma_i' \) is a valid signature, \( U'_i = u'_iP \) and \( W'_i = (d'_i \phi_i + h'_i x'_i + u'_i)Z \). At last, \( C \) outputs \( \sigma'_i = (U'_i, W'_i) \) to \( A_2 \), and as the signature on the last \( (\text{PID}_{ij}, m_i || TS_i) \), such a signature is valid.
of user $PID_t'$ on $m'_t||TS'_t$. If $PID'_t = PID_t$, $PK'_t = (r'_tP, aP)$, and $Z = bP$. Subsequently, $C$ performs the following process to solve CDHP:

- Compute $\phi'_i = H_4(y_{pub}, PK'_j, U'_t)$ for $(i = 1, 2, \ldots, n)$.
- Search $r'_t$ in the list $L_{ij}$, and calculate $k'_r = H_2(PID'_{ij}, R'_t)$ and $d'_r = r'_t + k'_rs$ for $i \neq t$. 
- Calculate $W'_i = (d'_r \phi'_i + h'_i x'_i + u'_i)Z$ for $i \neq t$.
- Calculate $W'_t = W'_t - \sum_{i=1, i \neq t} W'_i$, so $W'_t = ((r'_t + k'_rs)\phi'_t + h'_t a + u'_t)bP$.
- Search $x'_t$ in the list $L_{ij}$, and calculate $k'_t = H_2(PID'_{ij}, R'_t)$.
- Therefore, $C$ calculates $abP = k'^{-1}_t(W'_t - (u'_t + (r'_t + k'_rs)\phi'_t)bP)$ to resolve the CDLP.

6.3. Informal Security Analysis

We will informally analyze the improved CLAS scheme to satisfy security demands in the VANETs’ environment.

1. Authentication: Authentication can be achieved by the proof of Theorem 1. In Probabilistic Polynomial Time (PPT), no attacker can forge a valid signature. The verifier confirms the authenticity of the message and the validity of the signature by executing the Single Signature Verification or Aggregate Verification algorithm.

2. Nonrepudiation: In our CLAS scheme, TRA can recover its real identity $ID_i$ according to the vehicle’s pseudonym $PID_{ij}$, and the vehicle cannot deny the signature $\sigma_i$ generated by itself. Therefore, the proposed scheme supports nonrepudiation.

3. Anonymity: In VANETs, vehicles can only use the pseudonym $PID_i$ when communicating with other entities. When a vehicle wants to join VANETs, TRA runs the Pseudonym Generation algorithm to assign a pseudonym to the vehicle: $PID_{i, j} = ID_i \oplus H_3(kP + T_{ij})$, $PID_{i,j} = \{PID_{i, 1, j}, T_{ij}\}$. The authentic identity $ID_i$ of the vehicle is concealed within the pseudonym $PID_{i, j}$.

4. Unlinkability: The authentic identity $ID_i$ of the vehicle is hidden in the fake identity $PID_{i, j} = \{PID_{i, 1, j}, T_{ij}\}$ in this scheme, where $T_{ij} = t_{ij}P$, $PID_{i, 3, j} = ID_i \oplus H_1(kP + T_{ij})$. When transmitting different messages, the random numbers $t_{ij}$ ensure that the vehicle generates a different pseudonym each time. The attacker cannot associate two signatures to reveal the vehicle’s authentic identity, since their pseudonyms are only used once.

5. Traceability: When communicating with other vehicles and the RSU, the vehicle uses the pseudonym $PID_{i, j} = \{PID_{i, 1, j}, T_{ij}\}$. TRA tracks the authentic identity of the vehicle by computing $ID_i = PID_{i, 3, j} \oplus H_2(kP + T_{ij})$. The tracking key $k'$ is securely maintained by TRA. Consequently, in the event of a malicious incident involving a vehicle, only TRA possesses the capability to unveil the authentic identity of the vehicle.

6. Anti-replay attacks: In the improved CLAS scheme, when running the Signature Generation algorithm, each signature $\sigma_i$ contains a current time stamp $TS_i$. The verifier can verify the timeliness of the timestamp $TS_i$ to verify whether the message $m_i$ was replayed. Therefore, no one can replay the signed messages.

7. Anti-impersonation attack: If an attacker attempts to forge the vehicle’s pseudonym and send a fake message, the signature generated by the adversary will be rejected by the Single Signature Verification or Aggregate Verification mechanism. Thus, our proposed CLAS scheme supports a defense against impersonation attacks.

7. Performance Evaluation

In this section, we take a comparative analysis of the improved CLAS scheme with several CLAS schemes [21–23,28,37], encompassing factors such as computational overhead and communication overhead.
7.1. Computation Overhead

Simulation experiments comparing computational overhead were performed on a desktop consisting of an Intel(R) Core(TM) i5-11300H processor with 3.11 GHz of clock frequency and 16 GB of RAM, using Java to implement pairing-based cryptographic computations; referenced libraries include Java.security and it.unisa.dia.gas.jpbc. Table 3 shows some cryptographic symbols and execution times of corresponding cryptographic operations. We mainly calculate the computational burden of three parts of the scheme, as follows.

1. The vehicle generates the signature.
2. The RSU performs individual signature verification.
3. The TMC verifies the aggregated signature.

In this scheme, vehicles need to perform two elliptic curve scalar multiplications and two hash functions when generating a signature. When RSU validates a single signature, two bilinear pairing operations, three elliptic curve scalar multiplication operations, two elliptic curve scalar addition operations, and three hash functions are required. When verifying an aggregate signature, TMC needs to perform two bilinear pairing operations, \(3n\) elliptic curve scalar multiplication operations, \(3n\) elliptic curve scalar addition operations, and \(3n\) hash functions. In addition, the calculation overhead of other schemes can also be calculated according to this method. Table 4 provides a comparative analysis of the computational burdens associated with other schemes. In the end, Figure 2 shows the computational costs associated with generating and verifying a single signature. It is apparent that our scheme has the lowest cost of generating a single signature than others[21–23,28,37]. Moreover, the cost of verifying a single signature is less than other scheme [21,22,28,37] and slightly more than Wang et al. [23]. Further, the computational costs are shown in relation to the number of signatures in Figure 3.

![Figure 2. Computation overhead of signing and verifying one signature.](image)
Figure 3. The relationship between aggregation verification and the number of signatures.

Table 3. Execution time of cryptographic operations.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
<th>Running Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{bp}$</td>
<td>A bilinear pairing operation</td>
<td>10.3239</td>
</tr>
<tr>
<td>$T_{mtp}$</td>
<td>A map-to-point hash function of the bilinear pairing</td>
<td>2.4507</td>
</tr>
<tr>
<td>$T_{mul}$</td>
<td>A scalar multiplication operation of the bilinear pairing</td>
<td>1.1508</td>
</tr>
<tr>
<td>$T_{pa}$</td>
<td>A point addition operation of the bilinear pairing</td>
<td>0.0276</td>
</tr>
<tr>
<td>$T_{h}$</td>
<td>A one-way hash function</td>
<td>0.0015</td>
</tr>
</tbody>
</table>

Table 4. Computation overhead comparison.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Signature Generation (ms)</th>
<th>Single Signature Verification (ms)</th>
<th>Aggregate Verification (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhao et al. [21]</td>
<td>$2T_{mtp} + 4T_{mul} + 2T_{pa} + 2T_{h} \approx 9.5628$</td>
<td>$4T_{bp} + 2T_{mtp} + 2T_{mul} + T_{pa} + 2T_{h} \approx 48.5292$</td>
<td>$4T_{bp} + 2T_{mtp} + 2nT_{mul} + (4n - 3)T_{pa} + 2T_{h} \approx 46.1442$</td>
</tr>
<tr>
<td>Xu et al. [22]</td>
<td>$T_{mtp} + 3T_{mul} + T_{pa} + 2T_{h} \approx 7.7237$</td>
<td>$3T_{bp} + 2T_{mtp} + 2T_{mul} + T_{pa} + 2T_{h} \approx 39.3953$</td>
<td>$3T_{bp} + nT_{mtp} + 2nT_{mul} + (3n - 2)T_{pa} + 2T_{h} \approx 30.9165$</td>
</tr>
<tr>
<td>Wang et al. [23]</td>
<td>$3T_{mul} + T_{pa} + T_{h} \approx 3.4815$</td>
<td>$2T_{bp} + 2T_{mul} + 3T_{pa} + 2T_{h} \approx 23.0352$</td>
<td>$2T_{bp} + 2nT_{mul} + 3T_{pa} + 2nT_{h} \approx 33.3396$</td>
</tr>
<tr>
<td>Kamil et al. [26]</td>
<td>$T_{mtp} + 4T_{mul} + 2T_{pa} + 2T_{h} \approx 7.1121$</td>
<td>$3T_{bp} + 2T_{mtp} + 2T_{mul} + T_{pa} + 2T_{h} \approx 38.2053$</td>
<td>$3T_{bp} + (n + 1)T_{mtp} + 2nT_{mul} + (2n - 1)T_{pa} + nT_{h} \approx 3.7133n + 49.3278$</td>
</tr>
<tr>
<td>Altaf et al. [37]</td>
<td>$T_{mtp} + 2T_{mul} + T_{pa} + T_{h} \approx 4.7814$</td>
<td>$3T_{bp} + 2T_{mtp} + T_{mul} + T_{pa} + T_{h} \approx 37.053$</td>
<td>$3T_{bp} + (n + 1)T_{mtp} + nT_{mul} + (4n - 3)T_{pa} + nT_{h} \approx 3.7133n + 33.3396$</td>
</tr>
<tr>
<td>ours</td>
<td>$2T_{mul} + 2T_{h} \approx 2.3046$</td>
<td>$2T_{bp} + 3T_{mul} + 3T_{pa} + 3T_{h} \approx 24.1875$</td>
<td>$2T_{bp} + 3nT_{mul} + 3nT_{pa} + 3nT_{h} \approx 3.5397n + 20.6478$</td>
</tr>
</tbody>
</table>

7.2. Communication Overhead

We assess the communication burden of the enhanced scheme as well as several CLAS schemes. Given that the scheme relies on bilinear pairings, various parameters come into play, including the curve type within the bilinear pairing group, group order, and element length considerations. Specifically, the value of $p$ amounts to 64 bytes, while the elements of $G_1$ are sized at 128 bytes. Also, the sizes of the hash function output and
the timestamp are 20 bytes and 4 bytes, respectively. We assume that after receiving \( n \) signatures, RSU transmits an aggregated signature. For the convenience of calculation and comparison, we assume \( n = 100 \) for the analysis. Subsequently, Table 5 summarizes the comprehensive evaluation of the communication overhead. Furthermore, it is evident that the communication overhead of this approach is less than that of other schemes [21–23,37], and equal to Kamil et al. [28] as illustrated in Figure 4. But, the signature generation and verification cost of Kamil et al. [28] is higher.

Table 5. Communication overhead comparison.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Single Signature (Bytes)</th>
<th>Aggregate Signature (Bytes, ( n = 100 ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhao et al. [21]</td>
<td>2</td>
<td>(</td>
</tr>
<tr>
<td>Xu et al. [22]</td>
<td>2</td>
<td>(</td>
</tr>
<tr>
<td>Wang et al. [23]</td>
<td>3</td>
<td>(</td>
</tr>
<tr>
<td>Kamil et al. [28]</td>
<td>2</td>
<td>(</td>
</tr>
<tr>
<td>Altaf et al. [37]</td>
<td>2</td>
<td>(</td>
</tr>
<tr>
<td>ours</td>
<td>2</td>
<td>(</td>
</tr>
</tbody>
</table>

Figure 4. Aggressive signature size.

7.3. Practicality Assessment

In order to assess the processing capability of RSU, we introduce the RSU service capacity denoted as the \( Rsc \), and its calculation formula is [23]:

\[ Rsc = \frac{p \cdot d}{T_{ver} \cdot N \cdot v} \]

\( T_{ver} \) represents the duration needed for a single signature verification, which is 34.0827 ms. We make \( N \) denote the vehicle volume within 800 m of the RSU coverage. Meanwhile, \( v \) depicts the vehicle’s average speed, ranging from 5 to 20 m per second. Furthermore, \( p \) denotes the probability of a valid signature, and \( d \) corresponds to the distance of RSU coverage’s communication, assuming 1000 m. It is obvious from Figure 5 that \( Rsc \) gradually decreases as the vehicle density and velocity escalate. Therefore, a better \( Rsc \) of the RSU service capacity can be obtained by reducing the vehicle density.
Figure 5. Rsc in the improved scheme.

8. Conclusions

In this paper, we perform a security assessment of Wang et al.’s proposed CLAS scheme focusing on its conditional privacy-preserving in VANETs, and show that the scheme exhibits vulnerabilities to the KGC attack and public key replacement attack. Therefore, we present an enhanced CLAS scheme designed to fix the security issues. The security proof shows that the improved CLAS scheme effectively guards against type I and type II attackers within the standard model. It also realizes several security requirements specific to VANETs. Lastly, we assess the improved scheme’s performance with regard to computational cost and communication cost.

Author Contributions: Conceptualization, B.Y. and C.W.; methodology, B.Y. and C.W.; writing—original draft preparation, B.Y.; writing—review and editing, H.H. and C.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (Nos. 62372256, 61772292), the Natural Science Foundation of Fujian Province (Nos. 2023J01920, 2020J01905), the presidential research fund of Minnan Normal University (No. KJ18024) and the Science and Technology Project of Putian City (Nos. 2021R4001-10, 2022SZ3001ptxy05).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References


**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.