Hindawi
Security and Communication Networks
Volume 2021, Article ID 5049766, 17 pages
https://doi.org/10.1155/2021/5049766

Research Article

A Batch Authentication Design to Protect Conditional Privacy in Internet of Vehicles

Yuhao Yang,[1] **Xiujie Huang**,[1,2] **and Jinyu Hu**[1]

[1]*The College of Information Science and Technology, Jinan University, Guangzhou 510632, China*
[2]*The Guandong Key Laboratory of Data Security and Privacy Preserving, Guangzhou 510632, China*

Correspondence should be addressed to Xiujie Huang; t_xiujie@jnu.edu.cn

Received 3 September 2021; Revised 10 November 2021; Accepted 12 November 2021; Published 30 December 2021

Academic Editor: Azees M

Internet of vehicles (IoV), a novel technology, holds paramount importance within the transportation domain due to its ability to increase traffic efficiency and safety. Information privacy is of vital importance in IoV when sharing information among vehicles. However, due to the openness of the communication network, information sharing is vulnerable to potential attacks, such as impersonation, modification, side-channel and replay attacks, and so on. In order to resolve the aforementioned problem, we present a conditional privacy-preserving batch authentication (CPPBA) scheme based on elliptic curve cryptography (ECC). The proposed scheme avoids the certificate management problem, conducing to efficiency improvement. When a message is transmitted by a vehicle, its pseudo identity rather than the real identity is also broadcasted along with the shared message, which protects the privacy of the vehicle's identity. But this privacy is conditional because TA and only the TA can reveal the real identity of the vehicle by tracing. The proposed scheme is batch verifiable, which reduces the computation costs. In addition, our scheme does not involve bilinear pairing operations and does not use the map-to-point hash function, thus making the verification process more effective. An exhaustive efficiency comparison has been carried to show that the proposed CPPBA scheme has lower computation, communication, and storage overheads than the state-of-the-art ones. A relatively comprehensive security analysis has also been carried, which not only shows that the signature design in the CPPBA scheme is unforgeable under the random oracle model but also illustrates that the CPPBA scheme is resistant to various potential attacks. The security is also verified by a popular automated simulation tool, that is, AVISPA.

1. Introduction

With the rapid growth of networks and information technology, the internet of vehicles (IoV) has attracted more and more attention because of its ability to provide communication between vehicles, road side units (RSUs), and other devices (including personal devices and sensors), known as vehicle-to-everything (V2X) [1]. The vehicles ad hoc network (VANET), as a predecessor of IoV, effectively combines the driver, vehicle, and roads so as to provide the driver with information about the state of other vehicles outside the visual range [2], road conditions [3], and location-related life services [4], which is helpful to improve road safety and traffic efficiency. The VANET is equipped with wireless communication equipment road side units distributed along both sides of the road, which have sufficient energy supply, good wireless communication capabilities, and strong computing storage capabilities, and can also bear part of the computing overhead for the vehicle nodes. However, the VANET is a network that treats each vehicle node as a router and a custom node. The network coverage of these nodes is small, and the computing power is limited. Moreover, in the VANET, the network capacity is limited, and the wireless channel quality is unstable because it is affected by many factors, such as interference signals, complex infrastructure, and relative vehicle speed [5]. The IoV, as an evolution of the conventional VANET, is expected to remove these restrictions and promises huge commercial interest and research value [6].

To realize V2X communication, various protocols including the IEEE 802.11p standard, the dedicated short-range communication (DSRC), and cellular wireless communication are employed in IoV. Vehicles in IoV, equipped with an on-board unit (OBU), are usually connected by DSRC. Through GPS, radio frequency identification (RFID), sensors, cameras, image processing equipment, and so on, the vehicle can collect information about its own environment and surrounding vehicle status (e.g., road status, weather, and driving directions) and then broadcast the information within the coverage of the RSU. The RSU collects this information and rebroadcasts it along with other services or warning messages to other vehicles [7] (e.g., it is expected that safety information will be broadcasted every 100–300 milliseconds [8]). This information can be analyzed and processed by the OBU to provide the driver with a safe driving environment and plan the optimal driving route [9]. It can be seen that vehicles in IoV act as information providers and consumers at the same time. Therefore, it is very important to successfully realize effective data distribution in IoV applications.

When the information is shared in the IoV network, it is necessary to ensure that the right information reaches the right places at the right time. However, due to the openness of the communication network, hackers, malicious vehicles, and other nodes may change the information transmitted in IoV and pretend a legitimate node to send the bogus information over the IoV network. This results that the information sharing is vulnerable to potential attacks [10], such as side-channel attacks, impersonation attacks, modification attacks, and replay attacks. In order to resist various malicious attacks, one solution is to design a secure authentication scheme. Moreover, the environment in the IoV always changes because the vehicle communication range is short and the speed of the vehicle is high (over 36 km/h) [11]. This rapid change of the IoV network topology limits the communication time between vehicles. In addition, when in a traffic-intensive area, information exchange is very frequent, and the vehicle is usually required to be able to quickly authenticate a large amount of traffic-related information. In particular, when the shared information is a collision warning or emergency notification, this information is related to the safety of life and property, and a quick response must be made. To overcome the limitations of the computing and communication capabilities of the IoV, it is necessary to ask that the designed secure authentication scheme is efficient enough with a very low overhead in computation, communication, and storage.

Although extensive research of the authentication design has been conducted on the IoV in recent years, it has not yet been fully commercialized, partly because the above-mentioned challenges of security and efficiency still need to be resolved. (A short review of the related work on authentication designs for IoV will be given in the next section.) Therefore, a new CPPBA scheme is proposed. The main contributions of this article are summarized as follows:

(1) The proposed scheme is conditional privacy-preserving. That is, the vehicle uses a pseudo identity to share information, but TA and only the TA can reveal the real identity of the vehicle by tracing.

(2) The proposed scheme for IoV does not use the expensive pairing operations and map-to-point hash function so that it improves the computational efficiency of the system.

(3) The proposed scheme is proven secure and unforgeable against the potential adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$ under the ECDLP assumption, and the AVISPA tool is used to simulate the proposed scheme.

(4) Compared with the state-of-the-art schemes, our scheme has lower computation, communication, and storage overhead.

The remainder of this article is organized as follows. In Section 2, we present related work on various authentication schemes in VANET and IoV. System model and preliminaries are introduced in Section 3. Based on the model, our scheme is proposed in detail in Section 4. The security proof and analysis of the proposed scheme are presented in Section 5. In Section 6, the performance analysis of our scheme is shown to demonstrate its efficiency. Finally, we conclude this article in Section 7.

2. Related Work

There are mainly two types of authentication designs in IoV: one is on the basis of public-key infrastructure (PKI) and the other is on the basis of identity-based (ID-based) cryptography. The PKI-based authentication scheme has a disadvantage of low efficiency mainly because it requires a certificate authority (CA) to manage the identity/public key of the vehicle, which increases the computation, communication, and storage burdens. For example, in the anonymous certificate authentication scheme proposed by Raya and Hubaux in 2007 [12], it is necessary to install many anonymous certificates and public and private keys for each vehicle in advance, which brings a huge certificate management burden to CA. In the pseudonymous authentication scheme, called PASS, proposed by Sun et al. in 2010 [13], the size of the certificate revocation list (CRL) is linear with the number of revoked vehicles, and the vehicles are allowed to update certificates on road by the aid of RSU, which reduces the certificate management overhead to an acceptable extent at that time.

To improve efficiency, many researchers have invested a lot of energy to design ID-based authentication schemes because they exploit ID-based signatures and do not need to use certificates for identity verification. In order to protect the privacy of the user's identity, ID-based authentication schemes use pseudo identity to communicate over the IoV network. The ID-based authentication schemes can be roughly divided into two categories, namely, the bilinear-pairing-based authentication schemes those adopt bilinear pairings operations (e.g., see [14–17]), and the elliptic-curve-based authentication schemes are designed on the basis of the elliptic curve cryptography (ECC) without bilinear pairings (such as references [18–22]).

In 2016, to solve the problem of privacy leakage caused by the CRL checking process, Jiang et al. [14] proposed an anonymous batch authentication scheme using bilinear pairings, called ABAH, where the CRL checking process is performed by calculating a hash message authentication code. But the ABAH scheme [14] is inefficient due to the frequent update and revocation of the group key under the group key agreement and the complicated operations of bilinear pairings in processing batch authentication. In 2017, based on short-time anonymous certificates, Azees et al. [15] presented an authentication scheme using bilinear pairings, called EAAP. But the EAAP scheme does not protect against the bogus message attack, framing attack, Sybil attack, and replay attack [23]. In 2020, Ali and Li [16] proposed a scheme using bilinear pairings to achieve V2I secure communication. This scheme supports batch verification of messages to improve the efficiency of message verification, but compared with other schemes using elliptic curve construction, this scheme still has a larger computational and communication overhead. In 2021, Mei et al. [17] proposed an effective certificateless aggregation signature scheme. However, it suffers from the modification attack as proposed by Liu et al. [24], and has a huge computational and communication overhead because it uses bilinear pairings and map-to-point hash operations.

It is well-known that the bilinear pairings are the most expensive operation among all the cryptographic operations because of their highest computation and storage overhead. It is also well-known that ECC can provide higher security with smaller keys in size and then improve the computational and communication efficiency, storage capacity, and bandwidth efficiency. Particularly, ECC with 160 bit keys provides the same level of security as RSA with 1,024 bit keys [25]. Therefore, researchers are dedicated to studying the use of ECC to implement an identity-based signature scheme without bilinear pairings. In 2017, in order to solve the existing schemes that rely heavily on a tamper-proof device (TPD) or cannot satisfy the security requirement, Cui et al. [18] proposed the SPACF scheme that is based on software without relying on any special hardware. In the SPACF scheme [18], the Cuckoo filter and the binary search methods are presented to improve performance in the batch verification phase, but the communication interactions are very frequent such that it is less efficient in practical applications. Contrary to [18], in 2018, Xie et al. [19] proposed an efficient message authentication scheme supporting batch messages verification as well as signatures aggregation. In their scheme, the master key of the system will be loaded into the TPD of each registered vehicle. However, this approach is very dangerous because it means that the TPD of each vehicle contains the master key of the system. When the TPD of a certain vehicle is compromised, the entire authentication system will collapse. As we know, such an attack on TPD exists. That is, some attackers can obtain part of TPD information through side-channel attacks [26, 27], such as power analysis and laser scanning. In addition, it is more troublesome to revoke the identity of the vehicle because the master key of the system is already loaded in the TPD of each vehicle. Therefore, this approach is not advisable, and few researchers have continued this method afterwards. In the same year, Gayathri et al. [20] proposed a certificateless authentication scheme without pairings, thereby improving communication and computing efficiency. However, in Gayathri et al.'s scheme, a vehicle can impersonate other vehicles through the pseudonym generated by itself, and the secure key update is not provided, so it is vulnerable to impersonate attack and side-channel attack. In 2020, Sutrala et al. [21] proposed a conditional privacy protection certification scheme with pairings. However, we found that the scheme was not secure because the signature was forgeable and a forgery of the signature was constructed [28]. In 2021, Thumbur et al. [22] proposed an efficient authentication scheme based on signature aggregation. Similar to Mei et al.'s scheme [17], Thumbur et al.'s scheme is also vulnerable to modification attacks [24]. Finally, we summarize the related work on the authentication schemes for IoV in Table 1, which shows each design with the basis of ECC or bilinears and the property of batch authentication.

From the above review, it can be seen that the above authentication schemes for IoV still have some problems in security and efficiency. In order to fill these problems, we propose a secure and efficient ID-based conditional privacy protection batch authentication scheme on the basis of ECC without pairings. It can be shown that our scheme meets the security requirements, including anonymous authentication, message integrity, traceability, unlinkability, and so on and that our scheme has low computation, communication, and storage cost.

3. System Model and Preliminaries

In this section, the system model, security model, security goals, and the elliptic curve discrete logarithm assumption are briefly introduced.

3.1. System Model. As shown in Figure 1, the system model of the IoV in this paper consists of four entities: the trusted authority (TA), the key generation center (KGC), road side units (RSUs) fixed at the road side, and vehicles.

(1) TA: The TA is considered to be a fully trusted third party, which can be a government entity or a trusted organization entity with sufficient calculation and storage capabilities. It is responsible for generating system parameters and the registration of vehicles. Moreover, TA is the only entity that can track the vehicle's real identity when it maliciously spreads false information.

(2) KGC: The KGC is another trusted entity enriched with computation and communication resources in the system. The KGC is responsible for generating the pseudo identity and partial private key of the vehicle, and KGC and TA are two independent third parties.

(3) RSUs: RSUs are wireless communication equipment, such as base stations deployed along the road. The RSU acts as an intermediate node between the TA/

Table 1: A review of authentication schemes.

Scheme	PKI/ID-based	ECC/bilinear-based	Batch authentication
Raya et al. [12]	PKI	ECC	No
Sun et al. [13]	PKI	Bilinear-based	No
Jiang et al. [14]	ID-based	Bilinear-based	Yes
Azees et al. [15]	ID-based	Bilinear-based	Yes
Ali et al. [16]	ID-based	Bilinear-based	Yes
Mei et al. [17]	ID-based	Bilinear-based	Yes
Cui et al. [18]	ID-based	ECC	Yes
Xie et al. [19]	ID-based	ECC	Yes
Gayathri et al. [20]	ID-based	ECC	Yes
Sutra et al. [21]	ID-based	ECC	Yes
Thumbur et al. [22]	ID-based	ECC	Yes

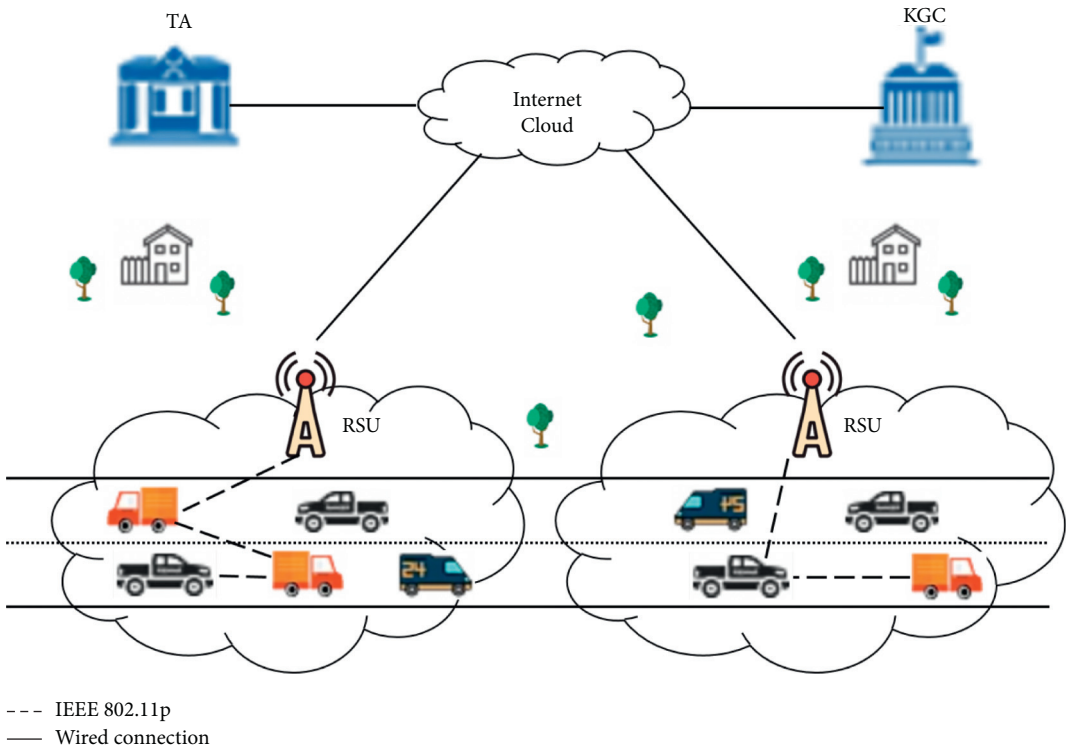

Figure 1: System model.

KGC and the vehicle. The RSU connects with the TA/KGC by a wired link and the vehicle by a wireless channel under the DSRC protocol (IEEE 802.11p) [29] or the cellular communication technology. The RSU can inspect the validity of the messages received from the vehicle and forward the valid message to nearby vehicles and RSUs.

(4) Vehicles: Each vehicle is equipped with an OBU, which is responsible for message generation and transmission to nearby vehicles and RSUs, and stores sensitive information in its TPD.

3.2. Security Model of CPPBA Scheme. In this subsection, the security model of CPPBA schemes will be described. Similar to [17, 21, 22, 30–32], we consider the following types of adversaries:

(1) Type-I adversary $\mathcal{A}_1$: This adversary is also called a public-key replacement attacker, who can compromise the vehicle's secret value or is capable to replace the public key of any vehicle with a value of his choice but cannot access the KGC's master secret key.

(2) Type-II adversary $\mathcal{A}_2$: This adversary is also called a malicious KGC attacker, who can access the KGC's master secret key but cannot replace the public key of any vehicle.

To show the capabilities of adversaries, the following two games, that is, Game I and Game II, are introduced, which are, respectively, performed by a Type-I adversary and a Type-II adversary interaction with some challengers.

Game I: This game is executed between the challenger Γ_1 and an adversary $\mathcal{A}_1$ as follows:

(1) Setup phase: In this phase, challenger Γ_1 initializes system parameters *params* and k, and then Γ_1 forwards *params* to the adversary $\mathscr{A}_1$ and keeps k secretly.

(2) H_3_query: The adversary $\mathscr{A}_1$ makes H_3_query on identity PID_i, and then challenger Γ_1 returns f_i to $\mathscr{A}_1$.

(3) H_4_query: The adversary $\mathscr{A}_1$ makes H_4_query on identity PID_i and message M, and then challenger Γ_1 returns h_i to $\mathscr{A}_1$.

(4) Sign_query: The adversary $\mathscr{A}_1$ makes signature queries on messages M under identity PID_i that are adaptively chosen by the adversary himself. Then, the challenger Γ_1 runs the signing algorithm to compute the signature σ_i to the adversary.

(5) Forgery: The adversary outputs a forged signature σ_i^* on message M under identity PID_i^* and wins the game if:

(a) σ_i^* is a valid signature on the message M under identity PID_i^*.
(b) There exists the case $PID_i^* = PID_i$ in the valid signature forged by $\mathscr{A}_1$.
(c) When the adversary $\mathscr{A}_1$ makes H_3_query on identity $PID_i^* = PID_i$, challenger Γ_1 returns $f_i^* \neq f_i$.

Game II: This game is executed between the challenger Γ_2 and an adversary $\mathscr{A}_2$ as follows:

(1) Setup phase: In this phase, challenger Γ_2 initializes system parameters *params* and k, and then Γ_2 forwards *params* and k to the adversary $\mathscr{A}_2$.

(2) H_3_query: The adversary $\mathscr{A}_2$ makes H_3_query on identity PID_i, and then challenger Γ_2 returns f_i to $\mathscr{A}_2$.

(3) H_4_query: The adversary $\mathscr{A}_2$ makes H_4_query on identity PID_i and message M, and then challenger Γ_2 returns h_i to $\mathscr{A}_2$.

(4) Extract_query: The adversary $\mathscr{A}_2$ makes Extract_query for PID_{i1} and the partial private key psk_{V_i} using PID_i; Γ_2 runs the Extract_query algorithm to compute the PID_{i1} and the partial private key psk_{V_i} and returns them to adversary $\mathscr{A}_2$.

(5) Sign_query: The adversary $\mathscr{A}_2$ makes signature queries on messages M under identity PID_i that are adaptively chosen by the adversary himself. Then, the challenger Γ_2 runs the signing algorithm to compute the signature σ_i to the adversary.

(6) Forgery: The adversary outputs a forged signature σ_i^* on message M under identity PID_i^* and wins the game if

(a) σ_i^* is a valid signature on the message M under identity PID_i^*,
(b) there exists the case $PID_i^* = PID_i$ in the valid signature forged by $\mathscr{A}_2$,
(c) when the adversary $\mathscr{A}_2$ makes H_4_query on identity $PID_i^* = PID_i$, challenger Γ_2 returns $h_i^* \neq h_i$

Definition 1: An authentication scheme in IoV is said to be provable security (i.e., existential unforgeability) if there are no polynomial-time Type-I and Type-II adversaries who can, respectively, win Game I and Game II with non-negligible advantages.

3.3. Security Goals. According to [16, 17, 21–23], we find that a secure authentication scheme in IoV should satisfy the following security requirements:

(1) Message authentication and integrity: The receiver should be able to verify the signature and inspect whether the received message was modified or forged.

(2) Conditional privacy-preserving or traceability: The vehicle's real identity should be hidden during message transmission and authentication processes to prevent the leakage of the vehicle's sensitive information. Only TA is able to track the vehicle's real identity from its signature.

(3) Unthinkability: No attacker can link any two received messages, even if two messages are sent from the same vehicle.

(4) Resistance to impersonation attack: In this type of attack, the attacker is able to imitate a legitimate vehicle to generate a valid signature. A secure CPPBA scheme should be able to prevent the impersonation attack.

(5) Resistance to message modification attack: In this type of attack, the attacker is able to modify the legitimate message that is transmitted over the network to achieve its specific purpose. For example, the attacker sends a fable traffic jam message to nearby vehicles to get a better traffic condition for itself. A secure CPPBA scheme should be able to prevent the message from modification attacks.

(6) Resistance to side-channel attack: In this type of attack, the attacker is able to attack the TPD of the vehicle by some physical methods to obtain part information stored in the TPD. In the secure CPPBA scheme, the secret stored in the TPD of the vehicle should not be disclosed by side-channel attacks.

(7) Resistance to replay attack: This attack is a type of network attack, in which some valid data is maliciously repeated or delayed in transmission. A secure CPPBA scheme should be able to withstand such an attack.

The security requirements of the CPPBA scheme will be analyzed in detail in Section 5.2.

3.4. Elliptic Curve Discrete Logarithm Assumption. Let p and q be two large prime numbers, and F_p represent a finite field of p elements. Suppose an elliptic curve is defined by the equation as follows: $y^2 = x^3 + ax + b \pmod{p}$, where $a, b \in F_p$ and $4a^3 + 27b^2 \pmod{p} \neq 0$. Let $\mathbb{O}$ be the point at infinity. The point $\mathbb{O}$ and all points on the elliptic curve form an additive group G of order q. Let P be a generator of group G. The scalar multiplication of the elliptic curve is defined as $kP = P + P \cdots + P\,(k\,times)$, where $k \in Z_q^*$.

Elliptic curve discrete logarithm problem (ECDLP) and assumption: Given two points on the elliptic curve $P, Q \in G$, where $Q = xP$ and $x \in Z_q^*$, the ECDLP problem is to determine the integer x. It is assumed that the ECDLP problem is hard when q is large.

4. The Proposed Scheme

For the requirement of conditional privacy-preserving and high authentication efficiency in IoV, a CPPBA scheme is proposed. Table 2 describes the notations used in our scheme. The details of the proposed scheme are described as follows, whose working flow is also illustrated in Figure 2.

4.1. System Initialization. This phase is performed by TA and KGC to generate the initial system parameters along with their public and private key pairs using the following steps:

(1) TA chooses two large primes p and q. TA selects elliptic curve additive group G of order q, which is defined by $\mathbb{E}$: $y^2 = x^3 + ax + b \pmod{p}$, where $a, b \in F_p$ and $4a^3 + 27b^2 \pmod{p} \neq 0$. P is a generator of the group G.

(2) TA randomly selects $s \in Z_q^*$ as its master key and computes $P_{\text{pub}} = sP$ as the corresponding public key.

(3) KGC randomly selects $k \in Z_q^*$ as its master private key and computes $K_{\text{pub}} = kP$ as the corresponding public key.

(4) TA chooses four one-way cryptographic hash functions H_1: $G \longrightarrow Z_q^*$, H_2: $\{0,1\}^* \longrightarrow Z_q^*$, H_3: $\{0,1\}^* \longrightarrow Z_q^*$, H_4: $\{0,1\}^* \longrightarrow Z_q^*$.

(5) Finally, TA and KGC broadcast public parameters $params = \{p, q, \mathbb{E}, P, G, P_{pub}, K_{pub}, H_1, H_2, H_3, H_4\}$.

4.2. Vehicle Registration. In this stage, TA communicates with the vehicle V_i in a secure channel. The following steps will be performed:

(1) V_i first sends its real identity RID_i to TA, which contains the real information of the user, such as the ID number of the vehicle owner, the license plate number, and vehicle identification number.

(2) TA computes $\alpha_i = H_1(RID_i)$ and stores the pair $\{RID_i, \alpha_i\}$. Then TA marks V_i as a registered vehicle and sends α_i to V_i via the secure channel.

4.3. Vehicle Partial Key and Pseudo Identity Generation. In order to protect the privacy of the vehicle, anonymous communication is necessary. On the other hand, the TA is asked to be able to reveal the real identity of the vehicle if necessary. Hence, this privacy is conditional. To this end, a pseudo identity is generated for each registered vehicle V_i along with the partial private key by KGC interaction with the vehicle through a secure channel before signing a message. This phase consists of the following steps:

(1) V_i sends α_i to KGC via the secure channel.

(2) KGC checks whether α_i exists in the registered vehicle list obtained from TA via the secure channel. If it does not exist, KGC terminates. Otherwise, KGC generates V_i's partial private key and pseudo identity by steps (3) and (4).

(3) KGC randomly chooses $\mu_i \in Z_q^*$ and computes $PID_{i1} = \mu_i P$ and $PID_{i2} = \alpha_i \oplus H_2(\mu_i P_{\text{pub}} \| T_{V_i})$. Let V_i's pseudo identity as $PID_i = \{PID_{i1}, PID_{i2}, T_{V_i}\}$, where T_{V_i} is the validity period of PID_i.

(4) Then, KGC computes V_i's partial private key as $psk_{V_i} = f_i k + \mu_i$, where $f_i = H_3(PID_i \| K_{pub})$.

(5) Finally, KGC sends $\{psk_{V_i}, PID_i\}$ to V_i via the secure channel. V_i stores $\{psk_{V_i}, PID_i\}$ in its TPD.

This phase is also shown in Figure 3.

4.4. Vehicle Key Generation and Message Signature. To ensure the integrity and validity of the message M_i, the vehicle signs the message M_i before broadcasting it.

(1) V_i randomly selects $r_i \in Z_q^*$ as its private key and computes $R_i = r_i P$.

(2) V_i computes $h_i = H_4(M_i \| PID_i \| T_i)$, where T_i is the current timestamp. Then V_i generates the signature $\sigma_i = \{\delta_i, R_i\}$, where $\delta_i = psk_{V_i} + r_i h_i$. Finally, V_i broadcasts the message-signature tuple $\langle M_i, PID_i, \sigma_i, T_i \rangle$ to the vicinal RSUs and other vehicles.

4.5. Signature Verification. When the RSU and other vehicles receive the message broadcasted by the vehicle V_i, the validity of the message is verified through the following steps:

(1) The receiver first validates T_{V_i} of PID_i and then checks the validity of the timestamp T_i. If $T_i' - T_i > \Delta T$, the receiver discards this message, where T_i' is the time when the receiver receives the message and ΔT represents the difference between the clock of the vehicle V_i and the local clock. Otherwise, the receiver continues to do the next verification.

Table 2: Notations and their meanings.

Symbol	Description
p, q	Two large primes
G	An additive group of elliptic curve points with prime order, q
P	A generator of G
$\mathbb{E}$	An elliptic curve
TA	Trusted authority
KGC	Key generation center
RSU	Road side unit
V_i	i^{th} vehicle
$PID_i = \{PID_{i1}, PID_{i2}, T_{V_i}\}$	V_i 's pseudo identity
T_{V_i}	Validity period of PID_i
RID_i	V_i 's real identity
(P_{pub}, s)	Public and private keys of TA
(K_{pub}, k)	Public and private keys of KGC
psk_{V_i}	V_i's partial private key
M_i	A message
σ_i	Signature on message, M_i
$H_1(), H_2(), H_3(), H_4()$	Secure one-way hash functions
$\|$	Concatenation operation
$\oplus$	Exclusive OR operation

TA	*KGC*		Vehicle V_i
Randomly choose $s \in Z_q^*$	Randomly choose $k \in Z_q^*$		
Compute $P_{pub} = sP$	Compute $K_{pub} = kP$		
	$\xleftarrow{RID_i}$		
Compute $\alpha_i = H_1(RID_i)$			
Store (α_i, RID_i)			
	$\xrightarrow{\alpha_i}$		
		$\xleftarrow{\alpha_i}$	
	Randomly choose $\mu_i \in Z_q^*$		
	Compute $PID_{i1} = \mu_i P$		
	$PID_{i2} = \alpha_i \oplus H_2(\mu_i P_{pub} \Vert T_{V_i})$		
	$f_i = H_3(PID_i \Vert K_{pub})$		
	$psk_{V_i} = f_i k + \mu_i$		
		$\xrightarrow{(psk_{V_i}, PID_i)}$	
			Randomly choose $r_i \in Z_q^*$
			Compute $R_i = r_i P$
			$h_i = H_4(M_i \Vert PID_i \Vert T_i)$
			Sign $\delta_i = psk_{V_i} + r_i h_i$
			$\sigma_i = (\delta_i, R_i)$
			Transmit $\langle M_i, PID_i, \sigma_i, T_i \rangle$

Figure 2: Working flow of our proposed scheme.

KGC	Vehicle V_i
	$\xleftarrow{\alpha_i}$
Randomly choose $\mu_i \in Z_q^*$	
Compute $PID_{i1} = \mu_i P$	
$PID_{i2} = \alpha_i \oplus H_2(\mu_i P_{pub} \Vert T_{V_i})$	
$f_i = H_3(PID_i \Vert K_{pub})$	
$psk_{V_i} = f_i k + \mu_i$	
	$\xrightarrow{(psk_{V_i}, PID_i)}$

Figure 3: Vehicle partial key and pseudo-identity generation.

(2) The receiver computes $f_i = H_3(PID_i \| K_{pub})$ and $h_i = H_4(M_i \| PID_i \| T_i)$ and checks whether equation (1) holds or not. If it holds, the receiver accepts the message. Otherwise, the receiver discards it.

$$\delta_i P = f_i K_{pub} + PID_{i1} + h_i R_i. \tag{1}$$

The correctness of equation (1) is proved as follows:

$$\begin{aligned} \delta_i P &= \left(psk_{V_i} + r_i h_i\right) P, \\ &= (f_i k + \mu_i + r_i h_i) P, \\ &= f_i K_{pub} + PID_{i1} + h_i R_i. \end{aligned} \tag{2}$$

4.6. Batch Verification. A vehicle may broadcast or receive multiple messages at the same time, especially in traffic-extensive areas. If the receiver verifies the messages one by one, it may cause redundant computation and a relatively long delay. In our scheme, efficient batch verification is developed, which is shown as follows:

(1) Suppose n messages-signature tuples $\{\langle M_1, PID_1, \sigma_1, T_1\rangle, \langle M_2, PID_2, \sigma_2, T_2\rangle, \ldots, \langle M_n, PID_n, \sigma_n, T_n\rangle\}$ are received. The receiver first checks the freshness of the timestamp T_i for message M_i and T_{v_i}. If T_i or T_{v_i} has expired, it rejects this message M_i. Otherwise, it performs the next verification.

(2) Let $\{\sigma_i\}_{i=1,\ldots,n'}$ be the list of signatures that are freshly generated having valid pseudo identities. The receiver selects a random vector $\overline{\lambda} = \{\lambda_1, \lambda_2, \ldots, \lambda_{n'}\}$, where $\lambda_i \in \{0,1\}^l$, l is usually 80 [24].

(3) The receiver computes $f_i = H_3(PID_i \| K_{\text{pub}})$ and $h_i = H_4(M_i \| PID_i \| T_i)$ and inspects whether the equation (3) holds. If it holds, the receiver accepts these n' messages.

$$\left(\sum_{i=1}^{n'} \lambda_i \delta_i\right) P = \left(\sum_{i=1}^{n'} \lambda_i f_i\right) K_{pub} + \sum_{i=1}^{n'} \lambda_i PID_{i1} + \sum_{i=1}^{n'} \lambda_i h_i R_i. \tag{3}$$

The correctness of equation (3) is as follows:

$$\begin{aligned} \left(\sum_{i=1}^{n'} \lambda_i \delta_i\right) P &= \sum_{i=1}^{n'} \lambda_i \left(psk_{V_i} + r_i h_i\right) P, \\ &= \sum_{i=1}^{n'} \lambda_i (f_i k + \mu_i + r_i h_i) P, \\ &= \left(\sum_{i=1}^{n'} \lambda_i f_i\right) K_{pub} + \sum_{i=1}^{n'} \lambda_i PID_{i1} + \sum_{i=1}^{n'} \lambda_i h_i R_i. \end{aligned} \tag{4}$$

5. Security Proof and Analysis

In this section, the security proof and analysis of our scheme proposed in Section 4 are given.

5.1. Security Proof. This subsection shows the provable security of our new scheme by using random oracle models under the Type-I (with Game I) adversary $\mathcal{A}_1$ and the Type-II (with Game II) adversary $\mathcal{A}_2$ as shown in Section 3.2.

Theorem 1. *If a polynomial adversary $\mathcal{A}_1$ can forge a valid signature by querying random oracles H_3_query, H_4_query, and Sign_query, then there exists a simulation algorithm Γ_1 that solves the ECDLP problem with non-negligible advantage.*

Proof. Suppose a polynomial adversary $\mathcal{A}_1$ can crack the proposed scheme with a non-negligible probability $\varepsilon > 0$. Our goal is to produce an algorithm Γ_1 that can use the adversary's ability to solve the ECDLP problem with non-negligible probability. That is, Γ_1 is able to compute k given two points P, $Q = kP \in G$. For this, Γ_1 takes PID^* as the target identity of $\mathcal{A}_1$ on a message M. The specific process is as follows:

Setup phase: Algorithm Γ_1 sets $K_{\text{pub}} = Q = kP$ and initializes system parameters params = $\{p, q, \mathbb{E}, P, G, P_{\text{pub}}, K_{\text{pub}}, H_1(\cdot), H_2(\cdot), H_3(\cdot), H_4(\cdot)\}$, and then Γ_1 forwards params to the adversary $\mathcal{A}_1$ and keeps k secretly. Γ_1 also maintains the lists H_3list and H_4list, which are initially empty.

H_3_query: When the adversary $\mathcal{A}_1$ makes H_3_query query on PID_i, Γ_1 checks the list H_3list for (PID_i, f_i). If such tuple exists, Γ_1 returns f_i to $\mathcal{A}_1$. Otherwise, Γ_1 randomly chooses $f_i \in Z_q^*$ and returns f_i to $\mathcal{A}_1$. Γ_1 adds the tuple (PID_i, f_i) to H_3list.

H_4_query: When the adversary $\mathcal{A}_1$ makes H_4_query query on (PID_i, M_i), Γ_1 checks the list H_4list for (PID_i, M_i, h_i). If such tuple exists, Γ_1 returns h_i to $\mathcal{A}_1$. Otherwise, Γ_1 randomly chooses $h_i \in Z_q^*$ and returns h_i to $\mathcal{A}_1$. Γ_1 adds the tuple (PID_i, M_i, h_i) to H_4list.

Sign_query: When $\mathcal{A}_1$ queries the Sign_query oracle using (PID_i, M_i). If $PID_i \neq PID^*$, Γ_1 aborts the process. If $PID_i = PID^*$, Γ_1 queries H_3_query and H_4_query oracles to obtain the tuples (PID_i, f_i) and (PID_i, M_i, h_i), respectively. Γ_1 randomly chooses $\delta_i, r_i \in Z_q^*$ and computes $R_i = r_i P$ and $PID_{i1} = \delta_i P - f_i K_{pub} - h_i R_i$, let $\sigma_i = \{\delta_i, R_i\}$. Next, Γ_1 sends $(\sigma_i = \{\delta_i, R_i\}, PID_i)$ to $\mathcal{A}_1$. The signature $\sigma_i = \{\delta_i, R_i\}$ is valid because σ_i satisfies equation (1) as follows:

$$\begin{aligned} \delta_i P &= f_i K_{\text{pub}} + PID_{i1} + h_i R_i, \\ &= f_i K_{\text{pub}} + \left(\delta_i P - f_i K_{\text{pub}} - h_i R_i\right) + h_i R_i, \\ &= \delta_i P. \end{aligned} \tag{5}$$

Forgery: $\mathcal{A}_1$ makes Sign_query query on (PID_i, M_i) to get the valid signature $\sigma_i = \{\delta_i, R_i\}$. By applying the forking lemma [33], when $PID_i^* = PID_i$, Γ_1 can obtain another valid signature $\sigma_i^* = \{\delta_i^*, R_i\}$ if it chooses different values in H_3_query random oracle with f_i^* by performing the same steps. Likewise, the signature is able to satisfy

$$\delta_i^* P = f_i^* K_{\text{pub}} + PID_{i1} + h_i R_i. \quad (6)$$

According to equations (5) and (6), we can deduce

$$(\delta_i - \delta_i^*)P = (f_i - f_i^*)kP. \quad (7)$$

Then by equation (7), the discrete logarithm can be computed $k = (\delta_i - \delta_i^*)/(f_i - f_i^*)$. Therefore, Γ_1 solves the ECDLP problem by outputting k.

The probability that Γ_1 resolves the ECDLP problem can be induced through the following events:

(1) $\mathscr{A}_1$ can forge a valid signature

(2) There exists the case $PID_i^* = PID_i$ in the valid signature forged by $\mathscr{A}_1$

(3) When the adversary $\mathscr{A}_1$ makes H_3_query on identity $PID_i^* = PID_i$, challenger Γ_1 returns $f_i^* \neq f_i$

Let q_s denotes the number of querying Sign_query oracle. So the probability that Γ_1 solves the ECDLP problem is at least $1/q_s(1 - 1/q_s)^{q_s - 1}\varepsilon$. And for large q_s, this probability turns to ε/eq_s, where e is the base of the natural logarithm. As a result, given the two points $P, Q = kP \in G$, Γ_1 can resolve the ECDLP problem with a non-negligible probability ε/eq_s, which causes a contradiction with the ECDLP assumption. □

Theorem 2. *If a polynomial adversary $\mathscr{A}_2$ can forge a valid signature by querying random oracles H_3_query, H_4_query,* Extract_query*, and Sign*_query*, then there exists a simulation algorithm Γ_2 that solves the ECDLP problem with non-negligible advantage.*

Proof. Suppose a polynomial adversary $\mathscr{A}_2$ can crack the proposed scheme with a non-negligible probability $\varepsilon > 0$. Our goal is to produce an algorithm Γ_2 that can use the adversary's ability to solve the ECDLP problem with non-negligible probability. That is, Γ_2 is able to compute x given two points P and $Q = xP \in G$. For this, Γ_2 takes PID^* as the target identity of $\mathscr{A}_2$ on a message M. The specific process is as follows:

Setup phase: Algorithm Γ_2 chooses $k \in Z_q^*$, computes $K_{pub} = kP$, and initializes system parameters params = $\{p, q, \mathbb{E}, P, G, P_{\text{pub}}, K_{\text{pub}}, H_1(\cdot), H_2(\cdot), H_3(\cdot), H_4(\cdot)\}$, and then Γ_2 forwards params and secret key k to the adversary $\mathscr{A}_2$. Γ_2 also maintains the lists H_3list, H_4list, and Exlist, which are initially empty.

H_3_query: When the adversary $\mathscr{A}_2$ makes H_3-query query on PID_i, Γ_2 checks the list H_3list for (PID_i, f_i). If such tuple exists, Γ_2 returns f_i to $\mathscr{A}_2$. Otherwise, Γ_2 randomly chooses $f_i \in Z_q^*$ and returns f_i to $\mathscr{A}_2$. Γ_2 adds the tuple (PID_i, f_i) to H_3list.

H_4_query: When the adversary $\mathscr{A}_2$ makes H_4-query query on (PID_i, M_i), Γ_2 checks the list H_4list for (PID_i, M_i, h_i). If such tuple exists, Γ_2 returns h_i to $\mathscr{A}_2$. Otherwise, Γ_2 randomly chooses $h_i \in Z_q^*$ and returns h_i to $\mathscr{A}_2$. Γ_2 adds the tuple (PID_i, M_i, h_i) to H_4list.

Extract_query: When $\mathscr{A}_2$ queries Extract_query for PID_{i1} and the partial private key psk_{V_i} using PID_i, Γ_2 inspects whether the tuple $(PID_i, PID_{i1}, psk_{V_i})$ exists in the list Exlist. If such tuple exists, Γ_2 forwards PID_{i1} and psk_{V_i} to $\mathscr{A}_2$. Otherwise, if $PID_i \neq PID^*$, Γ_2 makes H_3-query query to get the tuple (PID_i, f_i), then Γ_2 randomly chooses μ_i and computes $PID_{i1} = \mu_i P$ and $psk_{V_i} = f_i k + \mu_i$. Then Γ_2 returns PID_{i1} and psk_{V_i} to $\mathscr{A}_2$ and adds $(PID_i, PID_{i1}, psk_{V_i})$ to the list Exlist. If $PID_i = PID^*$, Γ_2 aborts the process.

Sign_query: When $\mathscr{A}_2$ uses (PID_i, M_i) to query the Sign_query oracle, if $PID_i \neq PID^*$, Γ_2 queries H_4-query and Extract_query oracles to obtain the tuple (PID_i, M_i, h_i) and $(PID_i, PID_{i1}, psk_{V_i})$, respectively. Γ_2 chooses a random number $r_i \in Z_q^*$ and computes $R_i = r_i P$ and $\delta_i = psk_{V_i} + r_i h_i$. Next, Γ_2 sends the signature $\sigma_i = \{\delta_i, R_i\}$ to $\mathscr{A}_2$. If $PID_i = PID^*$, Γ_2 queries H_3-query and H_4-query oracles to obtain the tuples (PID_i, f_i) and (PID_i, M_i, h_i), respectively. Γ_2 randomly chooses $\delta_i \in Z_q^*$ and computes $R_i = Q$ and $PID_{i1} = \delta_i P - f_i K_{pub} - h_i R_i$; let $\sigma_i = \{\delta_i, R_i\}$. Next, Γ_2 sends $(\sigma_i = \{\delta_i, R_i\}, PID_i)$ to $\mathscr{A}_2$. The signature $\sigma_i = \{\delta_i, R_i\}$ is valid because σ_i satisfies equation (1) as follows:

$$\begin{aligned} \delta_i P &= f_i K_{\text{pub}} + PID_{i1} + h_i R_i \\ &= f_i K_{\text{pub}} + \left(\delta_i P - f_i K_{\text{pub}} - h_i R_i\right) + h_i R_i \\ &= \delta_i P \end{aligned} \quad (8)$$

Forgery: $\mathscr{A}_2$ makes Sign_query query on (PID_i, M_i) to get the valid signature $\sigma_i = \{\delta_i, R_i\}$. By applying the forking lemma [33], when $PID_i^* = PID_i$, Γ_2 can obtain another valid signature $\sigma_i^* = \{\delta_i^*, R_i\}$ if it chooses different values in H_4-query random oracle with h_i^* by performing the same steps. Likewise, the signature is able to satisfy

$$\delta_i^* P = f_i K_{\text{pub}} + PID_{i1} + h_i^* R_i. \quad (9)$$

According to equations (8) and (9), we can deduce

$$(\delta_i - \delta_i^*)P = (h_i - h_i^*)xP. \quad (10)$$

Then by equation (10), the discrete logarithm can be computed $x = (\delta_i - \delta_i^*)/(h_i - h_i^*)$. Therefore, Γ_2 solves the ECDLP problem by outputting x.

The probability that Γ_2 resolves the ECDLP problem can be induced through the following events:

(1) $\mathscr{A}_2$ can forge a valid signature

(2) There exists the case $PID_i^* = PID_i$ in the valid signature forged by $\mathscr{A}_2$

(3) When the adversary $\mathscr{A}_2$ makes H_4-query on identity $PID_i^* = PID_i$, challenger Γ_2 returns $h_i^* \neq h_i$

Let q_E denotes the number of querying Extract_query oracle. So the probability that Γ_2 solves the ECDLP problem is at least $1/q_E(1 - 1/q_E)^{q_E - 1}\varepsilon$. And for large q_E, this

probability turns to ε/eq_E, where e is the base of the natural logarithm. As a result, given the two points $P, Q = xP \in G$, Γ_2 can resolve the ECDLP problem with a non-negligible probability ε/eq_E, which causes a contradiction with the ECDLP assumption. □

According to Definition 1 defined in Section 3.2, we can see, from the above two theorems, that our authentication scheme is existentially unforgeable (i.e., provably secure).

5.2. Security Analysis. In this subsection, the security analysis of the scheme proposed in Section 4 is discussed, which is similar to [16, 17, 21–23].

(1) Message authentication and integrity: When receiving message tuple $\langle M_i, PID_i, \sigma_i, T_i \rangle$, the receiver can verify message M_i through equation (1). Note that $h_i = H_4(M_i \| PID_i \| T_i)$ will change if the message M_i is modified, which will cause a failure of verification in equation (1). As a result, the proposed scheme can guarantee message authentication and integrity.

(2) Conditional privacy-preserving or traceability: Given the vehicle's pseudo identity $PID_i = \{PID_{i1}, PID_{i2}, T_{V_i}\}$, where $PID_{i1} = \mu_i P$ and $PID_{i2} = \alpha_i \oplus H_2(\mu_i P_{\text{pub}} \| T_{V_i})$, the pseudo identity of the vehicle does not contain the real information, so the attacker cannot obtain any real information from the vehicle's pseudo identity. However, once a legitimate vehicle deliberately spreads false information, TA can recover the real identity of the vehicle by computing $\alpha_i = PID_{i2} \oplus H_2(sPID_{i1} \| T_{V_i})$ and use α_i to query the data list to get the tuple $\{RID_i, \alpha_i\}$. Then TA adds RID_i to the revocation list.

(3) Unthinkability: In the proposed scheme, the vehicle's pseudo identity is $PID_i = \{PID_{i1}, PID_{i2}, T_{V_i}\}$, where $PID_{i1} = \mu_i P$ and $PID_{i2} = \alpha_i \oplus H_2(\mu_i P_{\text{pub}} \| T_{V_i})$, and the signature is $\delta_i = psk_{V_i} + r_i h_i$. Given the randomness of μ_i and r_i, it is impossible for an adversary to link any two pseudo identities PID_i and PID_j, or any two messages M_i and M_i' sent from the same vehicle V_i.

(4) Resistance to impersonation attack: According to Theorem 1, it is known that an attacker cannot impersonate other legitimate vehicles to generate a signature that satisfies equation (1). Hence, the proposed scheme can resist the impersonation attack.

(5) Resistance to message modification attack: In the signing phase, the vehicle V_i generates the signature $\delta_i = psk_{V_i} + r_i h_i$, where $h_i = H_4(M_i \| PID_i \| T_i)$ involves the hash value of the traffic-related message M_i. Once M_i is maliciously modified, the hash value h_i changes, thereby resulting in the failure of the verification of equation (1). Therefore, the proposed scheme can withstand the modification attack.

(6) Resistance to side-channel attack: In our scheme, the partial private key psk_{V_i} and the pseudo identity $PID_i = \{PID_{i1}, PID_{i2}, T_{V_i}\}$ of the vehicle is stored in its TPD, where T_{V_i} is the validity period of PID_i. When the vehicle's pseudo identity expires, the vehicle needs to reapply for a pseudo identity. KGC randomly chooses a new value μ_i', computes a new pseudo identity $PID_i' = \{PID_{i1}', PID_{i2}', T_{V_i}'\}$ and a new partial private key psk_{V_i}', where T_{V_i}' is the validity period of the new pseudo identity. Regular updates of the pseudo identity and the partial private key can effectively resist side-channel attacks.

(7) Resistance to replay attack: Timestamp T_i is involved in the message-signature tuple $\langle M_i, PID_i, \sigma_i, T_i \rangle$, where the signature is $\sigma_i = (\delta_i, R_i)$, $\delta_i = psk_{V_i} + r_i h_i$, and $h_i = H_4(M_i \| PID_i \| T_i)$. Hence, the receiver can detect whether the message M_i has expired by verifying the freshness of T_i. Accordingly, the proposed scheme is able to withstand the replay attack.

5.3. Simulation Result Analysis. In this subsection, we use the popular AVISPA tool to simulate the proposed scheme, and the simulation results are shown in Figure 4. The parts in the output format are as follows:

(1) SUMMARY: This part indicates whether the scheme is secure (safe or unsafe) or if the analysis was inconclusive

(2) DETAILS: This part provides information on the conditions in which the scheme is safe or the attack determining conditions or finally, why the analysis was inconclusive

(3) PROTOCOL: This defines the "HLPSL specification of the target protocol in IF"

(4) GOAL: It is the goal of the analysis performed by AVISPA using the HLPSL specification

(5)BACKEND: It is the name of the backend used for analysis

(6) STATISTICS: This is to track possible loopholes of the target protocol and statistics of some related data

The proposed scheme is simulated for formal security verification using the OFMC and CL-AtSe backends under the SPAN, the security protocol for AVISPA [34]. From Figure 4, it is clear that our scheme has passed the executability checking on non-trivial HLPSL specifications, replay attack checking, and Dolev–Yao model checking verifications [35]. Hence, our scheme is secure against replay and man-in-the-middle attacks.

6. Performance Analysis

In this section, we evaluate our scheme from the aspects of computation, communication, storage costs, and security performance by a comprehensive comparison with the state-of-the-art schemes proposed by Gayathri et al. [20] in

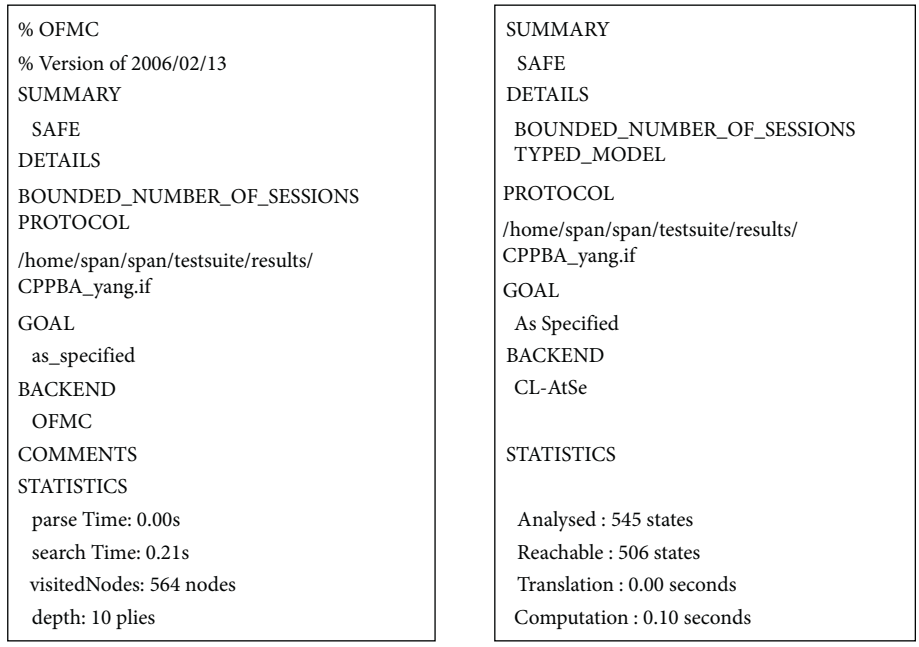

Figure 4: Analysis of simulation results under OFMC and CL-AtSe backends.

2018, Sutrala et al. [21] in 2020, and Mei et al. [17] in 2021. Among these three schemes, Mei et al.'s scheme uses bilinear pairings, while the other two schemes are pairing-free.

6.1. Computation Cost. The symbols used in the calculation and comparison with other schemes are listed below:

(1) T_{bp}: The execution time of bilinear pairing operation e: $G_1 \times G_1 \longrightarrow G_T$

(2) T_{pm}: The execution time of the scale multiplication in G_1

(3) T_{pa}: The execution time of a point addition operation in G_1

(4) T_{em}: The execution time of the scale multiplication operation kP in the elliptic curve, where $k \in Z_q^*$, $P \in G$

(5) T_{ea}: The execution time of the addition operation $P + Q$ in the elliptic curve, where $P, Q \in G$

(6) T_H: The execution time of a map-to-point hash function operation

(7) T_h: The execution time of a one-way hash function operation

In order to evaluate the time cost of the above cryptographic operations, we choose a Type A pairing that uses Java pairing-based cryptography (JPBC) library [36]. It is executed on a Dell desktop computer with the operating system being Windows 10 and the processor being CPU i7-9700, 8 GB RAM. We use the average time of 1,000 executions of the algorithm. The execution times of the above cryptographic operations are shown in Table 3.

In this paper, we adopt a similar evaluation method of computation costs as proposed in reference [37]. Let PKM, SMV, and BMV, respectively, represent the phase of pseudo identity generation, private key generation and message-signature generation, the phase of single message verification, and the phase of batch message verification. We only consider the vehicle's calculation time because the OBU's computing power is limited.

In our scheme, the vehicle needs to perform one multiplication operation and one one-way hash function operation in the PKM stage, which requires $T_{em} + T_h = 8.0962$ ms in time. In the SMV stage, three multiplication operations, two addition operations, and two one-way hash function operations are needed to be performed, which requires $3T_{em} + 2T_{ea} + 2T_h = 24.3608$ ms in time. In the BMV stage, $(2n + 2)$ multiplication operations, $2n$ addition operations, and $2n$ one-way hash function operations are needed to be performed when n messages are verified at one time, which requires $(2n + 2)T_{em} + 2nT_{ea} + 2nT_h = 16.2668n + 16.188$ ms in time. The above computation costs of our scheme are also shown in Table 4. For comparison, we also present the computation costs of those schemes proposed by Gayathri et al. [20], Sutrala et al. [21], and Mei et al. [17] in Table 4. According to the conclusion put forward by Liu et al. [24], in Mei et al.'s scheme, n messages are authenticated at a time by checking the equation $e(P, \sum_{i=1}^{n} \lambda_i T_i) = e(MPK, \sum_{i=1}^{n} \lambda_i h_i \quad Q_i) e(\sum_{i=1}^{n} \lambda_i h_i vepk_i, I_s) e(\sum_{i=1}^{n} \lambda_i U_i, J_s)$, where $\overline{\lambda} = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ is a random vector as defined in Section 4.6. The computation cost is $4nT_{em} + (4n - 4)T_{ea} + nT_h + 4T_{bp} + 2T_H = 32.527n + 113.843$ ms in time.

The percentage improvements of our scheme with respect to the related schemes are listed in Table 7. For instance, it has the improvement of $(24.2886 - 8.0962)/24.2886 \approx 66.67\%$, $(40.5882 - 24.3608)/40.5882 \approx 39.98\%$, and $[(24.4002n + 16.188) - (16.2668n + 16.188)]/(24.4002n + 16.188) \approx 33.11\%$ in the PKM, SMV, and BMV

Table 3: The cryptographic operations and execution times.

Cryptographic operation	Abbr.	Time (ms)
Operations related to bilinear pairing	T_{bp}	19.9421
	T_{pm}	7.7849
	T_{pa}	0.0405
Operations related to ECC	T_{em}	8.0940
	T_{ea}	0.0372
Map-to-point hash function	T_H	17.1117
One-way hash function	T_h	0.0022

phases, respectively, over the Gayathri et al.'s scheme, where $n = 100$ is the number of signatures. The percentage improvements in the PKM, SMV, and BMV phases over other related schemes can be calculated in a similar manner and are also shown in Table 5. From Table 5, it can be clearly seen that our scheme outperforms much better than these three schemes in terms of computational efficiency.

The computation costs in the PKM and SMV stages of different schemes are also represented graphically in Figure 5. From Figure 5, we can see easily that in the PKM and SMV phase, our scheme is much more efficient than Gayathri et al.'s scheme, Sutrala et al.'s scheme, and Mei et al.'s scheme. The curves of computation costs in the BMV phase for different schemes for various numbers of messages are depicted in Figure 6, which shows that our scheme is much superior to the other three schemes in the BMV phase.

6.2. Communication Cost. In this subsection, we evaluate the communication costs of our scheme with that of Gayathri et al.'s scheme, Sutrala et al.'s scheme, and Mei et al.'s scheme. Since the security level provided by the 160 bit ECC is the same as that by the 1024 bit RSA public-key cryptosystem [25], 160 bit ECC is adopted for the comparison of communication costs. A point on the elliptic curve, usually denoted by (P_x, P_y), is of length 320 bits or 40 bytes, while a point in the group over which a bilinear pairing is defined is of length 128 bytes. Then an element in Z_q^* is of length 20 bytes. Moreover, the timestamp is supposed to be of length 4 bytes.

In our scheme, the vehicle sends a message-signature tuple $\langle M_i, PID_i = \{PID_{i1}, PID_{i2}, T_{V_i}\}, \sigma_i = (\delta_i, R_i), T_i\rangle$ to the verifier, where $\langle PID_{i1}, R_i\rangle$ are two elements in the elliptic curve group G, $\langle PID_{i2}, \delta_i\rangle$ are two elements in Z_q^*, and $\langle T_{V_i}, T_i\rangle$ are two timestamps. So the communication cost is $2 \times 40 + 2 \times 20 + 2 \times 4 = 128$ bytes as shown in Table 6. Here, the cost of the message M_i does not included, which is a common means used in the communication cost comparison. Table 6 also shows us the communication costs of sending a single message-signature tuple by the vehicle in Gayathri et al.'s scheme, Sutra et al.'s scheme, and Mei et al.'s scheme, which are 228, 228, and 668 bytes, respectively. The communication costs for these schemes are also compared graphically in Figure 7. From Table 6 and Figure 7, it is clear that the communication cost of our scheme is significantly less than the other three schemes.

6.3. Storage Cost. In this subsection, the comparison of the storage space required by the vehicle in the signature phase for different schemes is presented.

In our scheme, the vehicle V_i needs to store the secret α_i, the pseudo identity $PID_i = \{PID_{i1}, PID_{i2}, T_{V_i}\}$, and the partial private key psk_{V_i} into its memory. So the required storage cost is $40 + 3 \times 20 + 4 = 104$ bytes as shown in Table 7. Also, shown in Table 7 are the storage costs required for Gayathri et al.'s scheme, Sutrala et al.'s scheme, and Mei et al.'s scheme, which are 184, 244, and 536 bytes, respectively. From Table 7, it can be clearly seen that the vehicle of our scheme has the smallest storage space compared with the other three schemes.

6.4. Security Comparison. In this subsection, we compare the various security and functionality features of our scheme with that of Gayathri et al.'s scheme, Sutrala et al.'s scheme, and Mei et al.'s scheme. Let $SF_1, SF_2, SF_3, SF_4, SF_5, SF_6, SF_7$, and SF_8 represent the goals of 1 message authentication and integrity, 2 identity privacy-preservation, 3 traceability, 4 unthinkability, 5 modification attack resistance, 6 impersonate attack resistance, 7 side-channel attack resistance, and 8 replay attack resistance, respectively. The comparison of security goals is indicated in Table 8. The symbol $\surd$ denotes that the security goal is satisfied, and the symbol $\times$ denotes that the security goal is unsatisfied.

From Table 8, it can be seen that our scheme provides all the mentioned necessary security features and is not vulnerable to any known attacks. In Gayathri et al.'s scheme, the vehicle can impersonate other vehicles through the pseudonym generated by itself, and a secure key update is not provided, so it is vulnerable to the impersonate attack and the side-channel attack. Sutrala et al.'s scheme is vulnerable to the modification attack and the impersonate attack, and TA cannot track the vehicle [28]. In Mei et al.'s scheme, the modification attack is possible [24]. Overall, compared with the existing related solutions in Table 8, our scheme provides a better security performance.

Table 4: Comparison of computation cost.

Scheme	PKM	SMV	BMV
Gayathri et al. [20]	$3T_{em} + 3T_h = 24.2886$ ms	$5T_{em} + 3T_{ea} + 3T_h = 40.5882$ ms	$(3n+2)T_{em} + 3nT_{ea} + 3nT_h = 24.4002n + 16.188$ms
Sutrala et al. [21]	$6T_{em} + 3T_h = 48.5706$ ms	$3T_{em} + 2T_{ea} + 2T_h = 24.3608$ ms	$(3n+1)T_{em} + (3n-1)T_{ea} + 2nT_h = 24.398n + 8.0568$ms
Mei et al. [17]	$6T_{em} + 2T_{ea} + 2T_H + T_h = 82.864$ ms	$4T_{bp} + 2T_{em} + 2T_H + T_h = 130.182$ ms	$4nT_{em} + (4n-4)T_{ea} + nT_h + 4T_{bp} + 2T_H = 32.527n + 113.843$ms
Our	$T_{em} + T_h = 8.0962$ ms	$3T_{em} + 2T_{ea} + 2T_h = 24.3608$ ms	$(2n+2)T_{em} + 2nT_{ea} + 2nT_h = 16.2668n + 16.188$ms

Table 5: Percentage improvements of our scheme with respect to the related schemes.

Scheme	PKM	SMV	BMV
Gayathri et al. [20]	66.67%	39.98%	33.11%
Sutrala et al. [21]	83.33%	0	32.89%
Mei et al. [17]	70.69%	81.29%	51.20%

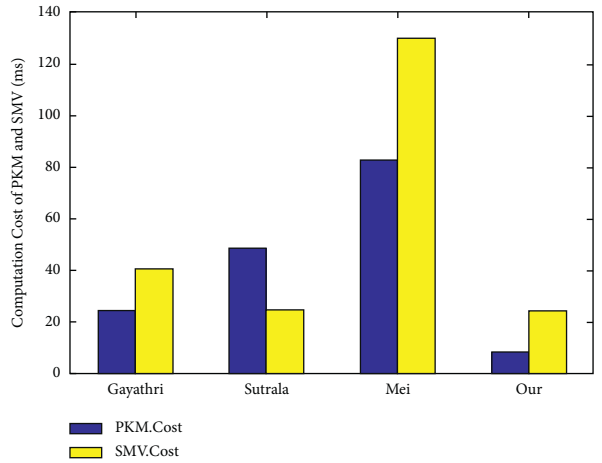

Figure 5: Comparison of PKM and SMV computation costs.

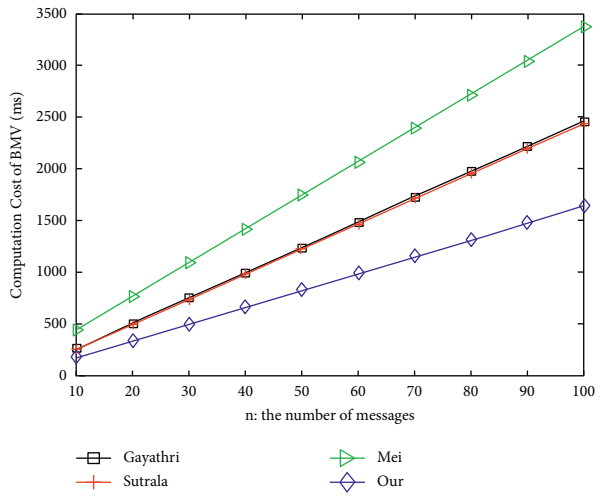

Figure 6: Comparison of BMV computation cost.

Table 6: Comparison of communication cost.

Scheme	Message	Sending a single message	Sending n message
Gayathri et al. [20]	$\langle M_i, PID_i = (PID_{i1}, PID_{i2}, T_{V_i}), X_i, \sigma_i = (R_i, Y_{1i}, \mu_i, w_i), T_i \rangle$	228 bytes	$228n$ bytes
Sutrala et al. [21]	$\langle M_i, \delta_i = \{f_i, g_i\}, B_i, K_i, R_i, PID_i = \{PID_{i1}, PID_{i2}, T_{V_i}\}, T_i \rangle$	228 bytes	$228n$ bytes
Mei et al. [17]	$\langle M_i, t_i, vepk_i, (PID_{i1}, PID_{i2}, T_{V_i}), ID_{R_s}, sig_i = (U_i, T_i) \rangle$	668 bytes	$668n$ bytes
Our	$\langle M_i, PID_i = \{PID_{i1}, PID_{i2}, T_{V_i}\}, \sigma_i = \{\delta_i, R_i\}, T_i \rangle$	128 bytes	$128n$ bytes

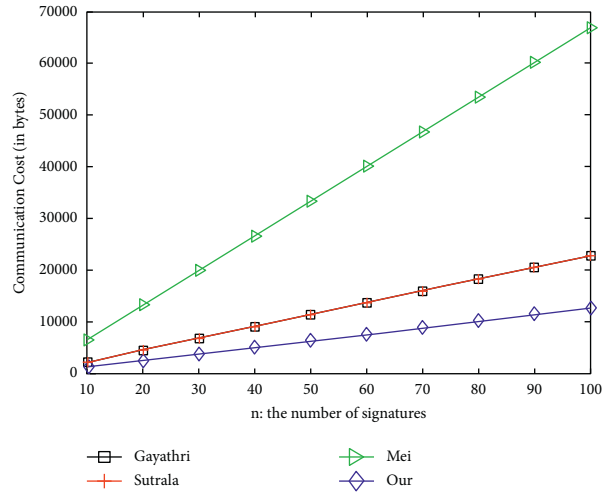

Figure 7: Comparison of communication cost for n signatures.

Table 7: Comparison of storage cost.

Scheme	Storage message	Storage cost
Gayathri et al. [20]	$\langle X_i, R_i, d_i, x_i, PID_i = (PID_{i1}, PID_{i2}, T_{V_i})\rangle$	184 bytes
Sutrala et al. [21]	$\langle vt_i, a_i, (PID_{i1}, PID_{i2}, T_{V_i}), (vpk_{1i}, vpk_{3i}, k_i), R_i, K_i\rangle$	244 bytes
Mei et al. [17]	$\langle (PID_{i1}, PID_{i2}, T_{V_i}), psk_{i,j}, x_i, vepk_i\rangle$	536 bytes
Our	$\langle \{PID_{i1}, PID_{i2}, T_{V_i}\}, \alpha_i, psk_{V_i}\rangle$	104 bytes

Table 8: Comparison of security.

Scheme	SF_1	SF_2	SF_3	SF_4	SF_5	SF_6	SF_7	SF_8
Gayathri et al. [20]	√	√	√	√	√	×	×	√
Sutrala et al. [21]	√	√	×	√	×	×	√	√
Mei et al. [17]	√	√	√	√	×	√	√	√
Our	√	√	√	√	√	√	√	√

7. Conclusion and Discussion

In this article, we present a secure and efficient conditional privacy-preserving batch authentication scheme based on elliptic curve cryptography. In our scheme, TA is responsible for vehicle registration and generates information α that is bound to the vehicle's real identity for the vehicle. Then the vehicle can request KGC to generate a pseudo identity and partial private key for it through message α. This procedure of pseudo identity and partial private key generation is renewed periodically as needed, which prevents side-channel attacks on the TPD of the vehicle. After the vehicle obtains the pseudo identity and partial private key, the vehicle generates the private key and uses the private key to sign the message and then broadcasts the signature together with its pseudo identity and the message. The identity privacy of the vehicle is preserved by broadcasting its pseudo identity rather than the real identity over the IoV network, and this privacy is conditional since any entity except the TA cannot reveal the real identity from the pseudo identity. When many messages are received simultaneously at the vehicle or the RSU, a procedure of batch verification can be conducted to reduce the computation. Our scheme is shown to be secure by proof of unforgeability for the signature and a comprehensive analysis of necessary security features and resistances to various potential attacks. The cost of our scheme in terms of computation, communication, and storage is exhaustive compared to several state-of-the-art schemes that demonstrates that the overall performance of our new scheme is better.

Our authentication scheme protects the vehicle's identity privacy and meets the necessary security features, but it, similar to previous designs [15–22], has some limitations. For example, it does not resist a collusion attack. That is, if a (large) group of vehicles collude and send a bogus message by the authentication scheme, the message will pass verification, and no collusive vehicle will be revealed. As the development of the internet of things and artificial intelligence, data poisoning attacks [38], under which a large number of users may be trapped to broadcast bogus traffic-related information over the IoV network, have attracted more and more attention, which will make the (automated) vehicle fail to schedule the best route. A solution to recover this limitation is to introduce the trust mechanism [39] into the cryptographic authentication approaches, which is our future work. In addition, motivated by the three-factor authentication and key agreement technology over the users/vehicles and the servers [40], the factor of biometrics

can also be introduced in the design of a CPPBA scheme to realize an effective connection between the vehicles and the users. Before the user logs in to the vehicle, the user will be recognized by its biometric, and only the legitimate user can access the vehicle. When in disputes and responsibilities, the TA can reveal not only the real identity of the vehicle but also the real identity of the user. This is also an interesting and important direction in the future.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the NSF of China under Grant no. 61871201, the Guangdong Provincial NSF under Grant no. 2021A1515011906, and the Teaching Reform Research Projects of Jinan University under Grant no. JG2020158.

References

[1] Y. Miao, X. Liu, K.-K. R. Choo, R. H. Deng, H. Wu, and H. Li, "Fair and Dynamic Data Sharing Framework in Cloud-Assisted Internet of Everything," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 7201–7212, 2019.

[2] X. Xiaodong Lin, R. Rongxing Lu, C. Chenxi Zhang, H. Haojin Zhu, P.-h. Pin-Han Ho, and X. Xuemin Shen, "Security in vehicular ad hoc networks," *IEEE Communications Magazine*, vol. 46, no. 4, pp. 88–95, 2008.

[3] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode, "TrafficView: traffic data dissemination using car-to-car communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, no. 3, pp. 6–19, 2004.

[4] U. Uichin Lee, J. Jiyeon Lee, J.-S. Joon-Sang Park, and M. Gerla, "FleaNet: A Virtual Market Place on Vehicular Networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 1, pp. 344–355, 2010.

[5] M. Jain and R. Saxena, "Overview of VANET: Requirements and its routing protocols," in *Proceedings of the 2017 International Conference on Communication and Signal Processing (ICCSP)*, pp. 1957–1961, IEEE, Chennai, India, April 2017.

[6] Y. Ni, J. He, L. Cai, J. Pan, and Y. Bo, "Joint Roadside Unit Deployment and Service Task Assignment for Internet of Vehicles (IoV)," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3271–3283, 2019.

[7] P. K. Singh, S. K. Nandi, and S. Nandi, "A tutorial survey on vehicular communication state of the art, and future research directions," *Vehicular Communications*, vol. 18, p. 100164, 2019.

[8] H. Noori and B. B. Olyaei, "A novel study on beaconing for VANET-based vehicle to vehicle communication: Probability of beacon delivery in realistic large-scale urban area using 802.11p," in *Proceedings of the 2013 International Conference on Smart Communications in Network Technologies (SaCoNeT)*, pp. 1–6, IEEE, Paris, France, June 2013.

[9] G. Singh, "Video streaming communication over VANET," in *Recent Advances in Computational Intelligence*, vol. 823, pp. 189–197, Springer, 2019.

[10] S. Kumar and K. S. Mann, "Prevention of DoS Attacks by Detection of Multiple Malicious Nodes in VANETs," in *Proceedings of the 2019 International Conference on Automation, Computational and Technology Management (ICACTM)*, pp. 89–94, IEEE, London, UK, April 2019.

[11] I. Ali, T. Lawrence, A. A. Omala, and F. Li, "An Efficient Hybrid Signcryption Scheme With Conditional Privacy-Preservation for Heterogeneous Vehicular Communication in VANETs," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 11266–11280, 2020.

[12] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of computer security*, vol. 15, no. 1, pp. 39–68, 2007.

[13] Y. Sun, R. Lu, X. Lin, X. Shen, and J. Su, "An Efficient Pseudonymous Authentication Scheme With Strong Privacy Preservation for Vehicular Communications," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 7, pp. 3589–3603, 2010.

[14] S. Jiang, X. Zhu, and L. Wang, "An Efficient Anonymous Batch Authentication Scheme Based on HMAC for VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 8, pp. 2193–2204, 2016.

[15] M. Azees, P. Vijayakumar, and L. J. Deboarh, "EAAP: Efficient Anonymous Authentication With Conditional Privacy-Preserving Scheme for Vehicular Ad Hoc Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2467–2476, 2017.

[16] I. Ali and F. Li, "An efficient conditional privacy-preserving authentication scheme for Vehicle-To-Infrastructure communication in VANETs," *Vehicular Communications*, vol. 22, p. 100228, 2020.

[17] Q. Mei, H. Xiong, J. Chen, M. Yang, S. Kumari, and M. K. Khan, "Efficient Certificateless Aggregate Signature With Conditional Privacy Preservation in IoV," *IEEE Systems Journal*, vol. 15, no. 1, pp. 245–256, 2021.

[18] J. Cui, J. Zhang, H. Zhong, and Y. Xu, "SPACF: A Secure Privacy-Preserving Authentication Scheme for VANET With Cuckoo Filter," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10283–10295, 2017.

[19] Y. Xie, F. Xu, D. Li, and Y. Nie, "Efficient message authentication scheme with conditional privacy-preserving and signature aggregation for vehicular cloud network," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–12, Article ID 1875489, 2018.

[20] N. B. Gayathri, G. Thumbur, P. V. Reddy, and M. Z. Ur Rahman, "Efficient Pairing-Free Certificateless Authentication Scheme With Batch Verification for Vehicular Ad-Hoc Networks," *IEEE Access*, vol. 6, pp. 31808–31819, 2018.

[21] A. K. Sutrala, P. Bagga, A. K. Das, N. Kumar, J. J. P. C. Rodrigues, and P. Lorenz, "On the Design of Conditional Privacy Preserving Batch Verification-Based Authentication Scheme for Internet of Vehicles Deployment," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5535–5548, 2020.

[22] G. Thumbur, G. S. Rao, P. V. Reddy, N. B. Gayathri, D. V. R. K. Reddy, and M. Padmavathamma, "Efficient and Secure Certificateless Aggregate Signature-Based Authentication Scheme for Vehicular Ad Hoc Networks," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1908–1920, 2021.

[23] J. Li, Y. Ji, K.-K. R. Choo, D. Hogrefe, and CL-CPPA, "CL-CPPA: Certificate-Less Conditional Privacy-Preserving

Authentication Protocol for the Internet of Vehicles," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10332–10343, 2019.

[24] J. K. Liu, T. H. Yuen, M. H. Au, and W. Susilo, "Improvements on an authentication scheme for vehicular sensor networks," *Expert Systems with Applications*, vol. 41, no. 5, pp. 2559–2564, 2014.

[25] N. Jansma and B. Arrendondo, "Performance comparison of elliptic curve and RSA digital signatures," *Efficiency Comparison of Elliptic Curve and RSA Signatures*, vol. 5.

[26] W. Cilio, M. Linder, C. Porter, J. Di, D. R. Thompson, and S. C. Smith, "Mitigating power- and timing-based side-channel attacks using dual-spacer dual-rail delay-insensitive asynchronous logic," *Microelectronics Journal*, vol. 44, no. 3, pp. 258–269, 2013.

[27] H. J. Mahanta, A. K. Azad, and A. K. Khan, "Differential Power Analysis: attacks and Resisting Techniques," in *Information Systems Design and Intelligent Applications*, J. K. Mandal, S. C. Satapathy, M. Kumar Sanyal, P. P. Sarkar, and A. Mukhopadhyay, Eds., Springer India, New Delhi, pp. 349–358, 2015.

[28] Y. Yang and X. Huang, *Comments on "On the Design of Conditional Privacy Preserving Batch Verification-Based Authentication Scheme for Internet of Vehicles Deployment", Cryptology ePrint Archive*, IEEE, Guangzhou, China, 2021.

[29] J. B. Kenney, "Dedicated Short-Range Communications (DSRC) Standards in the United States," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

[30] K.-A. Shim, "Security models for certificateless signature schemes revisited," *Information Sciences*, vol. 296, pp. 315–321, 2015.

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

[32] A. K. Malhi and S. Batra, "An efficient certificateless aggregate signature scheme for vehicular ad-hoc networks," *Discrete Mathematics & Theoretical Computer Science*, vol. 17, no. 1.

[33] D. Pointcheval and J. Stern, "Security Proofs for Signature Schemes," in *Advances in Cryptology — EUROCRYPT '96*, U. Maurer, Ed., Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 387–398, 1996.

[34] A. Armando, D. Basin, Y. Boichut et al., "The AVISPA tool for the automated validation of internet security protocols and applications," *Computer Aided Verification*, vol. 3576, pp. 281–285, 2005.

[35] C. Lv, M. Ma, H. Li, J. Ma, and Y. Zhang, "An novel three-party authenticated key exchange protocol using one-time key," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 498–503, 2013.

[36] Java Pairing-Based Cryptography Library (JPBC), http://gas.dia.unisa.it/projects/jpbc/docs/pairing.html.

[37] S.-J. Horng, S.-F. Tzeng, Y. Pan et al., "b-SPECS+: batch verification for secure pseudonymous authentication in VANET," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 11, pp. 1860–1875, 2013.

[38] J. Chen, X. Zhang, R. Zhang, C. Wang, L. Liu, and A. De-Pois, "De-Pois: An Attack-Agnostic Defense against Data Poisoning Attacks," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3412–3425, 2021.

[39] R. Hussain, J. Lee, and S. Zeadally, "Trust in VANET: A survey of current solutions and future research opportunities," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 2553–2571, 2021.

[40] Q. Jiang, N. Zhang, J. Ni, J. Ma, X. Ma, and K.-K. R. Choo, "Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9390–9401, 2020.