

## Research Article

# A Certificateless Anonymous Cross-Domain Authentication Scheme Assisted by Blockchain for Internet of Vehicles

Xueyan Liu <sup>1</sup>, Li Wang <sup>2</sup>, Linpeng Li <sup>1</sup>, Xiaoyan Zhang <sup>2</sup> and Shufen Niu <sup>1</sup>

<sup>1</sup>College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, China

<sup>2</sup>College of Mathematics and Statistics, Northwest Normal University, Lanzhou 730070, China

Correspondence should be addressed to Xueyan Liu; liuxy@nwnu.edu.cn

Received 1 July 2022; Revised 31 October 2022; Accepted 4 November 2022; Published 21 November 2022

Academic Editor: SK Hafizul Islam

Copyright © 2022 Xueyan Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the development of the Internet of Things and the increase of intelligent vehicles, the Internet of Vehicles (IoVs) have been widely used in the information communication such as road and traffic conditions. However, heavy overhead of certificate management, high computing load of identity and message authentication, and the privacy disclosure of vehicle nodes have hindered the development of intelligent transportation. In this study, we propose a certificateless cross-domain anonymous authentication scheme based on blockchain for IoVs. Specifically, the vehicle identity information is authenticated by the first roadside unit (RSU), and transactions are recorded permanently and immutably in the blockchain to reduce the repeated authentication load of other RSUs. To achieve conditional privacy, the trusted authority (TA) generates pseudonyms for each registered user. The relation between the pseudonym and the real identity is kept confidential by the TA and only can only be revealed in case of disputes. Meanwhile, the private key of the vehicle is generated anonymously on the basis of certificateless technology and the pairing-free signature verification. Correctness and security proof demonstrate that our proposed scheme is provably secure and can withstand different types of attacks. A simulation environment has been built to test the packet loss rate and delay of messages in the network. Results show that the proposed scheme is more efficient than the related schemes.

## 1. Introduction

With the rapid development and maturity of the Internet of Things, the Internet of Vehicles (IoVs), as the basic technology of intelligent transportation system, have received extensive attention and research from academia and the industry. IoVs provide a safer driving environment by allowing vehicles to communicate with one another or with roadside infrastructure to improve road safety, driving conditions, and comfort for road users.

IoVs are formed by the combination of vehicles and intelligent network equipment, which is composed of a high-speed mobile wireless ad hoc networks. Figure 1 shows the typical IoV environment, including trusted authority (TA), roadside units (RSUs), vehicles, and the Internet. The communication is carried out by vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I), in which valuable

driving information and exchange traffic-related information are shared to improve driving and passenger safety [1].

IoVs bring safety and provide various entertainment information services for drivers and passengers. However, at the same time, it will inevitably generate massive data and face many information security challenges. Firstly, given the openness and fragility of wireless networks, the messages transmitted in IoV system are vulnerable to various attacks, such as modification or impersonation. Moreover, the privacy of vehicles has also been greatly threatened, such as exposing user behaviour tracks. Secondly, the vehicles in IoVs move at high speed, and the communication bandwidth is minimal, which requires that vehicle identity and message authentication in IoVs to have lower computation and communication overhead. In addition, tens of thousands of vehicles and their onboard devices will realise the sharing and interaction of vehicle or driving data through

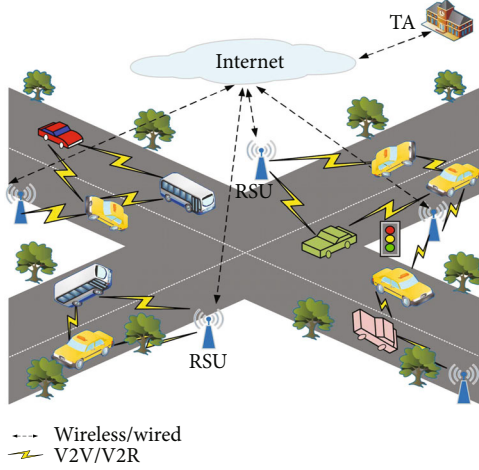


FIGURE 1: IoV network model.

the IoVs. This exchange results in massive data, which will inevitably cause high communication overhead and reduce the performance of each entity in the IoV system. Of course, it also leads to the increase of packet loss rate.

To solve the traffic-related message security problems in IoVs, researchers propose authentication schemes based on public key infrastructure (PKI), identity-based cryptography (IBC), and certificateless cryptography (CLC). In the schemes based on PKI, to manage users' public keys, a certification authority must store many certificates. Therefore, the RSUs require sufficient computation and storage capacity to validate these certificates. To alleviate the certificate management problem caused by PKI, IBC is introduced into the IoVs. In identity-based authentication schemes, the public key derives from the user's identity information, and the private key generator (PKG) has no certificate storage load. However, PKG can use the user's private key to forge the user's signature and thus cause a key escrow problem. To solve this problem, Al-Riyami and Paterson proposed CLC in 2003 [2]. Then, the researchers also proposed the corresponding CLC-based authentication schemes.

Some certificateless signature schemes use bilinear pairing operation with a large amount of computation; hence, they are unsuitable for IoV environment with low delay. Therefore, a signature scheme with low delay and minimal computation is urgently needed to improve the performance of all aspects in IoVs.

**1.1. Motivation and Contributions.** Zheng et al. [3] proposed a traceable distributed IoV system framework based on blockchain technology by adopting the authentication scheme between vehicles and RSUs. However, there are two problems in the authentication stage of this scheme:

**Problem 1.** When a vehicle reaches the range of each RSU on the road, it sends an authentication request to the RSU containing the pseudonym and the corresponding public key to achieve identity authentication by each RSU. This frequent and repeated operation cannot meet the low communication requirements and low computation requirements in the IoV system, which increases the system burden.

**Problem 2.** After determining the authenticity of the vehicle identity, the RSU initiates the random integer negotiation process and sends the vehicle a random integer encrypted by the vehicle's public key and stored in the on-board unit (OBU). However, this phase only considers the communication between the vehicle and the first RSU in a particular area. When the vehicle enters the next RSU area, this process will be repeated and the storage load of OBU will be increased.

Therefore, on the basis of the work of Zheng et al. [3], we propose a certificateless anonymous cross-domain authentication scheme assisted by blockchain, and its main contributions are as follows:

- (1) To preserve the privacy of vehicles, the TA distributes pseudonyms to each vehicle for the whole communication process. The key generation centre (KGC) generates partial private keys of the vehicle, and the vehicle combines partial private keys with its secret values to generate the actual private keys to solve the key escrow problem and void heavy certificate verification load
- (2) To overcome Problems 1 and 2, the vehicle sends the verification information calculated by itself and the pseudonym assigned by the TA to the first RSU, and only the first RSU negotiates a random integer with the vehicle. Then, the RSU packages the negotiated random integer and other contents on the blockchain built by the RSUs to realise cross-domain authentication and reduce the computational overhead caused by repeated signature authentication
- (3) The homomorphic signature without bilinear pairing is adopted to improve the efficiency of RSUs verifying messages
- (4) When a malicious event occurs, according to the openness and transparency of the blockchain, other vehicles can report the malicious event to RSUs. The RSUs will present the pseudoidentity of the malicious vehicle to TA. Finally, the TA traces the true identity from the user list, revokes the user, and updates the user list accordingly

**1.2. Organization.** This research is organized as follows. Section 2 presents the existing related work. Section 3 shows the background knowledge and describes the system model. Section 4 shows the basic construction. Security proof and performance evaluation will be given in Section 5 and Section 6. The last section makes a conclusion.

## 2. Related Works

Anonymous identity authentication is a typical method to preserve vehicle privacy in IoVs. Many researchers have studied and proposed privacy preservation authentication schemes in IoVs based on the basic idea of using digital pseudonyms as a unique identifier to authenticate without

any personal identity information. Firstly, in PKI-based scheme, certificate authorities need to generate multiple anonymous public-private key pairs and certificates for each OBU and prestore them in tamper-proof devices (TPD) [4, 5]. Therefore, it is difficult to manage and store a large number of public-private key pairs and related certificates, which also increases the complexity of maintenance and management.

To solve the above problems, scholars have designed a variety of IBC-based schemes. IBC was first launched by Shamir in 1984 [6]. It sets the entity's digital identity as a public key, eliminating the need for the key infrastructure, and KGC uses the primary key to generate the entity's private key. Although such schemes avoid PKI's certificate management problems, they often need to carry out time-consuming pairing operations [7] and introduce key escrow problems. Song et al. proposed a batch authentication scheme using elliptic curve bilinear pairs and pointed out the existing security risks. By improving the program of Song et al. [8], an identity-based batch verification security scheme [9] with bilinear pairing-free is proposed. However, in these identity-based schemes, the main problem is that KGC uses its master key to generate a key for a vehicle entity. It cannot ensure nonrepudiation because KGC abuses the vehicle's access ability to sign and decrypt any message, resulting in key escrow problems.

To address these problems of key escrow and certificate management, Al-Riyami and Paterson introduced a certificateless-based mechanism in 2003 [2]. Yao et al. [10] proposed a certificateless anonymous authentication mechanism named CLMA. The mechanism applies the key exchange technology supporting password authentication based on ring fault-tolerant learning problems, which can provide mutual authentication when vehicles access vehicle cloud services through RSUs. Xu et al. [11] proposed a certificateless authentication protocol named SE-CLASA, which does not rely on a fully trusted third party; the signature scheme supports aggregation, which can resist information injection attacks. Malhi and Batra [12] proposed a new certificateless aggregate signature scheme for VANETs with constant pairing computations. However, vehicles need to be registered repeatedly in different regional transportation authority management areas, and the amount of registration calculation is relatively high. In 2018, Wang and Teng [13] proposed a verifiable and secure certificateless aggregate signature algorithm in VANETs. However, Yang et al. [14] pointed out that the study by Wang and Teng [13] was not secure and proposed an improved certificateless aggregate signature scheme. Zhao and Zhang [15] proposed an authenticable privacy preservation scheme based on certificateless ring signcryption, but this scheme used a time-consuming bilinear pairing operation. Hathal et al. [16] presented a certificateless and lightweight authentication scheme. In their work, they introduced authentication tokens to reduce the burden of certificate management. However, the authentication of each vehicle node requires the participation of TA, so a bottleneck problem arises.

Yang et al. [17] proposed a method of using blockchain to protect the data privacy of IoVs. Although this method

is also based on certificateless cryptography system to realise the signature and authentication of blockchain transactions, combined with edge computing devices, it proposed a fine-grained data access control method with an efficient partially hidden access strategy. Ali et al. [18] also proposed a certificateless signature scheme based on blockchain, but many bilinear pairings are used in both schemes to verify the signature and are unsuitable for IoV systems with low delay and computational complexity. Bagga et al. [19] proposed blockchain-based batch authentication protocol for Internet of Vehicles, in which vehicles and RSUs can be added dynamically. Unfortunately, at the registration stage of vehicles and RSUs, TA generates certificates for them and a certificate management problem arises. In addition, the real identity, pseudoidentity, certificate, and part of private key of the vehicle and RSU are all generated by TA, so the calculation pressure of TA is large. Subsequently, Ren et al. [20] proposed an efficient and privacy-preserving certificateless public key signature scheme based on the blockchain. Although their scheme added two blockchains to the structure to protect the identity privacy of vehicles, it did not describe the construction of the blockchain network.

At present, many scholars have put forward a lot of other types of signature verification schemes, such as 6G-enabled VANETs [21] and anonymous signature-based authentication [22]. However, most of them have some problems, such as key management, high computational, and communication costs. Consequently, we utilize the certificateless cryptosystem to solve the key escrow problem, and a pairing-free authentication method is used for efficiency consideration. Furthermore, we introduce blockchain technology to achieve transparency and nontamperability of transaction information and reduce duplicate certification load.

### 3. Preliminaries

In this section, we will present cryptographic materials, system models, adversary models, design objectives, and symbols.

#### 3.1. Cryptography Materials

**3.1.1. Elliptic Curve.** Suppose that the symbol  $E/F_p$  denotes an elliptic curve  $E$  over a prime finite field  $F_p$ , where  $p$  is a large prime number. The curve  $E$  is defined as follows:

$$y^2 = x^3 + ax + b \pmod{p}. \quad (1)$$

Such that  $a, b \in F_p$  and  $\Delta = 4a^3 + 27b^2 \pmod{p} \neq 0$ . The points on  $E/F_p$  and a point at infinity  $O$  construct a cyclic additive group:

$$G = \{(x, y): x, y \in F_p, E(x, y) = 0\} \cup \{O\}. \quad (2)$$

A scalar multiplication over  $E/F_p$  can be computed as follows [23]:

$$tP = P + P + \dots + P \text{ (} t \text{ times)}, \quad (3)$$

in which  $t \in F_p$  and  $P \in G$ .

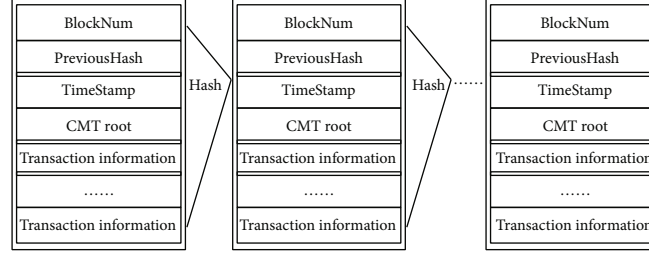


FIGURE 2: Blockchain structure.

**3.1.2. Elliptical Curve Discrete Logarithm Problem (ECDLP).**  $G$  is a finite cyclic group of prime order  $p$  defined on elliptic curve  $E$ ,  $pP$  is the generator of group  $G$ . Suppose that there is a random point  $Q$  in group  $G$ , and solve  $s$  so that it satisfies  $Q = sP$ .

**3.2. Certificateless Cryptosystem.** The concept of certificateless encryption and signature was first proposed by Al-Riyami and Paterson, which solves the issue of key escrow in IBC. The basic idea of certificateless encryption and signature is that KGC generates a partial private key for the user. The user's private key is jointly generated by the user and KGC. The user selects a secret value and combines it with the partial private key to generate the user's full private key. KGC does not know the user's complete private key, thus avoiding the problem of key escrow. The versatile definition of certificateless encryption and signature scheme consists of five algorithms, and the user public/secret key pair can be generated independently by the user even before obtaining the user partial key from the KGC [24]. Next, the definition of certificateless signature scheme will be presented:

- (i) Setup (master key generation): on input security parameter  $1^\lambda$ , it generates a master public/secret key pair (mpk, msk). The system parameter params is broadcasted to the other entities
- (ii) PartialKeyGen (user partial key generation): on input msk and user identity ID, it generates a user partial key  $d_{ID}$
- (iii) UserKeyGen (user key generation): on input mpk and user identity ID, it generates a user/private key pair  $(pk_{ID}, x_{ID})$
- (iv) CL\_Sign (signature generation): on input user private key  $x_{ID}$ , user partial key  $d_{ID}$ , and message  $m$ , it generates a signature  $\sigma$
- (v) CL\_Ver (signature verification): on input mpk, user identity ID, user public key  $pk_{ID}$ , message  $m$ , and signature  $\sigma$ , it returns 1 or 0 for accept or reject, respectively

Certificateless public key cryptography does not need certificate management and requires less load. Therefore, it is more suitable for mobile security applications with low broadband requirements and low energy consumption.

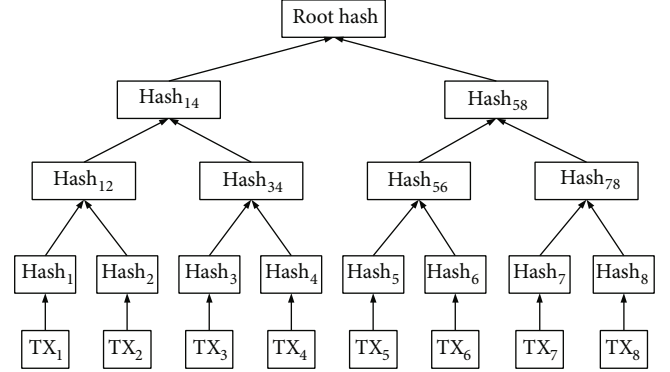


FIGURE 3: The structure of chronological Merkle tree.

**3.3. Blockchain.** Blockchain originates from a paper published by Nakamoto in 2008 [25]. It is a distributed ledger on a peer-to-peer network, with each node storing and backing up complete ledger information. As shown in Figure 2, it is a data structure that connects the generated data blocks sequentially in chronological order, a decentralized shared ledger that cannot be tampered and forged. Blockchain is an effective technology to deal with vehicle management and data transmission. A reasonable construction of vehicle blockchain network framework can effectively solve many problems in IoVs, such as broadcast conflict, resource scheduling, and privacy preservation. Therefore, promoting the deep integration of blockchain technology and IoVs is the inevitable trend of IoV development.

Each node user in the blockchain system can create a smart contract by publishing a transaction and use programming to set the smart contract as its own ownership transfer rules, transaction methods, and state transition functions. The blockchain verifies the correctness and integrity of the signature message data obtained by the RSUs through the deployed smart contract. When the data is correct, the smart contract is triggered to return the correct verification result. Otherwise, an error verification is returned.

The distributed ledger is encrypted using a Merkle tree. In this paper, we have only covered chronological Merkle tree (CMT). As shown in Figure 3, all transactions are hashed and stored chronologically in the CMT. Only the root hash is contained in the blockchain. In our proposed scenario, the transactions broadcast by RSUs are permanently recorded in the CMT, making the activities of each entity in the IoVs transparent and verifiable to the authorities.

**3.4. System Model.** Based on the framework of Zheng et al., we propose a certificateless anonymous cross-domain authentication system assisted by blockchain. Therefore, the entity is divided into five types: TA, KGC, cloud server (CS), RSUs, and vehicles (V). In the VANET system, the RSUs communicate with the TA through wired, while they communicate with the vehicles in its areas through wireless. This system has overcome two problems discussed in Section 1, and at the same time, with the participation of blockchain technology, the framework has the characteristics of reliability and security, reducing the dependence of traditional solutions on TA or KGC. The IoV network model is shown in Figure 4.

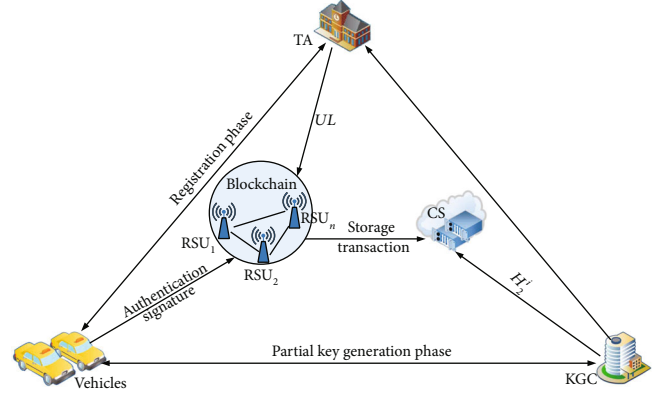


FIGURE 4: System model of the proposed scheme.

TABLE 1: Summary of notations.

Notation	Description
$V_i$	The $i$ th vehicle
$G$	An elliptic curve group
$\lambda$	Security parameter
$s$	TA's master secret key
$u$	KGC's master secret key
$T_{pub}$	TA's public key
$K_{pub}$	KGC's public key
$h_i$	A hash function
$RID_i$	$V_i$ real identity
$PID_i$	$V_i$ pseudoidentity generated by TA
$psk_i$	$V_i$ partial private key
$V_{pub_i}$	$V_i$ public key
$SK^i$	$V_i$ private key
$N_{R_j}$	An integer negotiated between the $V_i$ and $RSU_j$
$M_i$	A message generated by $V_i$
$\sigma_i$	A signature on $M_i$

- (1) TA: it is a fully trusted authority with unlimited computing resources and storage space. It is responsible for generating public/private keys and system parameters, registering vehicles, assigning pseudonyms to vehicles, and providing identity anonymity in vehicle communication
- (2) KGC: it is not fully trusted entity. It is responsible for building and allocating partial private keys for anonymous vehicles in IoVs
- (3) CS: it is responsible for managing the pseudonyms of vehicles issued by TA to facilitate identity verification and decentralized storage of transaction details including traffic information released by vehicles
- (4) RSUs: it is deployed on the roadside as a bridge between TA, KGC, and vehicles. All RSUs establish a blockchain network as peer nodes. The blockchain network is in charge of the collective maintenance of blockchain data and broadcasting messages to vehicles along the road. The first RSU of each area verifies the identity of vehicles through V2I communication and submits the verification result, which is recorded on the chain after verification. After receiving the signature message, each RSU is responsible for verifying the signature of message. It will be recorded in the blockchain after verification
- (5) Vehicle: the vehicle is equipped with an OBU to collect, calculate, and communicate traffic-related information. The device has its own clock to generate the correct timestamp. It is responsible for storing privacy information in the TPD of the vehicle and broadcasting information to the vehicle and nearby RSUs. In addition, vehicles can report malicious vehicles to RSUs

**3.5. Adversary Model.** Assume that TA and RSUs are trusted entities. The vehicle is equipped with TPD, so the adversary cannot read, write, or delete the contents of the TPD. This scheme is using the ideas of certificateless cryptography. Therefore, the user's public key has not been certificated. In the adversary model, adversaries have the right to replace the user's public key with the illegal public key chosen by themselves. Moreover, since KGC knows the system master key, which can calculate all part of the user private key,

but he cannot replace the user's public key. Therefore, we divide the adversary types into two categories. The adversary of type I simulates a malicious user who can request and replace the public key in the system. The adversary of type II is an internal attacker who has access to the KGC's master key and acts as a malicious but passive KGC.

**3.6. Notations.** The notations used in this paper are given in Table 1.

**3.7. Design Goals.** According to IoVs and practical application requirements, the new scheme shall meet the following properties:

- (1) Message authentication: it ensures that the received message has not been modified or forged in the process of communication
- (2) Anonymity: in this scheme, only TA can obtain the real identity of the vehicle. Other vehicles in IoVs and

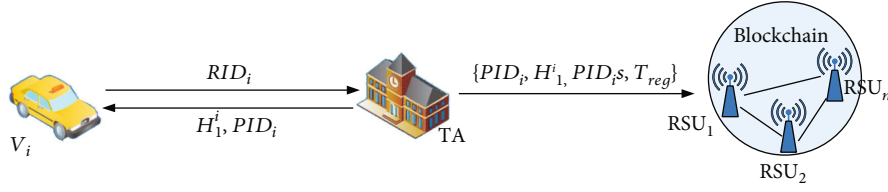


FIGURE 5: Distribution of registration data.

adversary cannot identify the real identity of the sender by analyzing multiple messages sent by the same vehicle

(3)Unlinkability: an attacker cannot find that multiple messages are from the same sender, and all pseudonyms should not reveal any connection between them

(4)Traceability and revocation: TA can track out the real identity of the vehicle sending malicious messages and revoke the malicious vehicle

(5)Unforgeability: any attacker cannot forge a legitimate signature

(6)Resilience to other attacks: blockchain-assisted certificateless anonymous authentication schemes should be able to withstand various common attacks in IoVs (such as impersonation, modification, replay, and man-in-the-middle attacks)

## 4. Concrete Scheme

In this section, we describe a concrete construction of our scheme, which consists of seven phases: system initialization phase, registration phase, partial key generation phase, key generation phase, identity authentication phase, message publishing and validation phase, and tracking.

**4.1. System Initialization Phase.** This phase is executed by TA and KGC which inputs a security parameter  $\lambda$  and generates parameters  $\{q, P, G\}$ .

(1)TA selects a master secret key  $s \in Z_q^*$  which is a random number and sets  $T_{pub} = sP$  as its master public key

(2)KGC selects  $u \in Z_q^*$  randomly as its master secret key and sets  $K_{pub} = uP$  as its master public key

(3)TA chooses five distinct cryptographic hash functions  $h_1 : \{0, 1\}^* \rightarrow Z_q^*$ ,  $h_2 : Z_q^* \times G \rightarrow \{0, 1\}^*$ ,  $h_3 : Z_q^* \rightarrow \{0, 1\}^*$ ,  $h_4 : G \times Z_q^* \rightarrow Z_q^*$ ,  $h_5 : Z_q^* \rightarrow \{0, 1\}^*$ .

(4)TA publishes system parameters  $params = \{G, P, T_{pub}, K_{pub}, h_1, h_2, h_3, h_4, h_5\}$ . The TA and KGC keep master secret keys  $s$  and  $u$  secretly, respectively

**4.2. Registration Phase.** Vehicle  $V_i$  with identity  $RID_i$  initiates a registration request. First, TA checks the existence of the real identity of the vehicle. If it exists,  $H_1^i = h_1(RID_i || T_{reg})$  is calculated, where  $T_{reg}$  is the registration time of the vehicle. Then, randomly select  $t \in Z_q^*$  and compute the pseudonym  $PID_i = tH_1^i \bmod q$ . Upload the pseudonym information  $(H_1^i, PID_i, PID_i s, T_{reg})$  to the user list UL which is put in the blockchain, and store a tuple  $(RID_i, H_1^i, PID_i, T_{reg})$  in TA's database. Finally,  $H_1^i, PID_i$  are sent to the vehicle. Figure 5 shows the registration process.

**4.3. Partial Key Generation Phase.** Vehicle  $V_i$  requests the partial key from KGC with its pseudonym  $PID_i$ . Once received  $PID_i$  from vehicle  $V_i$ , KGC works as follows:

(1)The KGC first looks up the user list UL obtained from blockchain to ensure that  $PID_i$  is present in UL, which means that  $PID_i$  has not been revoked by TA

(2)If  $PID_i$  is found in UL, KGC computes  $psk_i = u + PID_i \cdot s \bmod q$  and  $H_2^i = h_2(PID_i || psk_i^{-1}P)$

(3)KGC sets  $psk_i$  as the partial private key and sends  $H_2^i$  to the CS via the secure channel

**4.4. Key Generation Phase.** After receiving partial keys  $psk_i$  from KGC, vehicle  $V_i$  first verifies its correctness through the calculation equation  $e(psk_i P, P) = e(K_{pub} + PID_i T_{pub}, P)$ . Next, it selects  $x_i \in Z_q^*$  randomly and computes and publishes public key  $V_{pub_i} = x_i P$ , and the private key is  $SK^i = (x_i, psk_i)$ .

**4.5. Identity Authentication Phase.** During identity authentication, the vehicle  $V_i$  communicates with the RSUs by using its own pseudonym, and neither the RSUs nor the CS learns to acquire the vehicle's real identity. Figure 6 shows the identity authentication process.

- (1) When the vehicle  $V_i$  comes in the range of first  $RSU_j$  in a region, it sends an authentication request with its own  $PID_i$  and  $V_i$  computes  $psk_i^{-1}P$  and then sends it and  $PID_i$  to  $RSU_j$ . In order to determine whether the  $V_i$  is legal,  $RSU_j$  computes  $H_2^{i'} = h(PID_i || psk_i^{-1}P)$  and sends the result to CS, and CS queries whether it is equal to its stored  $H_2^i$ . If the vehicle is legal, CS sends the result back to  $RSU_j$ , and finally,  $RSU_j$  broadcasts the authentication result into blockchain network and which will be recorded on the blockchain after verification
- (2) After determining the authenticity of the vehicle identity, the  $RSU_j$  initiates the random integer negotiation process, which sends a random integer  $N_{R_j}$  encrypted by the vehicles' public key  $V_{pub_i}$  to the vehicle  $V_i$
- (3) The vehicle receives the ciphertext and decrypts it with its private key. After obtaining  $N_{R_j}$ , to determine whether the vehicle receives  $N_{R_j}$  from  $RSU_j$  and its integrity, the vehicle  $V_i$  has to select another random integer  $M_{V_i}$ , computes  $H_3^i = h_3(N_{R_j} || M_{V_i})$ , and sends  $H_3^i || M_{V_i}$  to  $RSU_j$ . To ensure the integrity

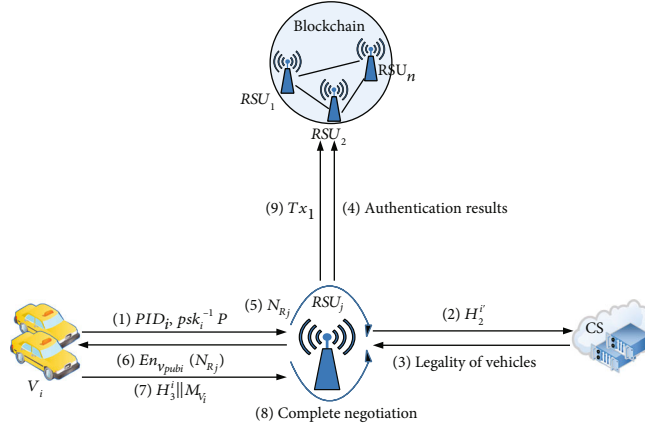


FIGURE 6: The identity authentication process.

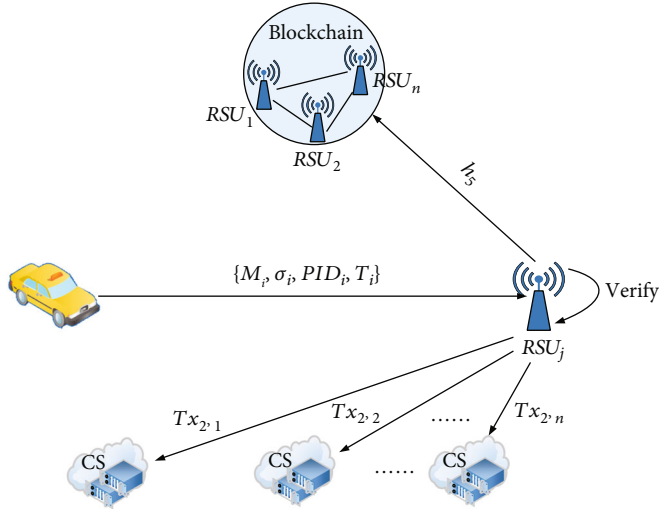


FIGURE 7: The message publishing and validation phase.

of the transaction information, it will perform  $h_3$  operation according to  $N_{R_j}$  and received  $M_{V_i}$ . If the result is consistent with the  $H_3^i$  computed by the vehicle  $V_i$ , an integer  $N_{R_j}$  is negotiated between the vehicle  $V_i$  and the  $RSU_j$ . Finally, the vehicle stores the  $N_{R_j}$  in OBU for later message publishing

- (4) In order to let the vehicle communicates conveniently with other RSUs, the  $RSU_j$  computes  $L_{ij} = M_{V_i}^{-1} N_{R_j}$  and packages  $PID_i$ ,  $V_{pub_i}$ , and  $L_{ij}$  into a transaction  $Tx_1 = (PID_i || V_{pub_i} || L_{ij})$  and then records on the blockchain indicating that  $N_{R_j}$  can be used by other RSUs in a certain region and the identity of vehicle  $V_i$  has been verified; that is, other RSUs do not need to repeatedly verify the identity of vehicle  $V_i$

**4.6. Message Publishing and Validation Phase.** After identity authentication phase, the vehicle  $V_i$  can publish message

and the  $RSU_j$  should verify them. Figure 7 is the scenario of message publishing and validation.

- (1) The vehicle  $V_i$  computes  $N_p = N_{R_j} P$  and  $H_4^i = h_4(N_p || M_i || PID_i || T_i)$  where  $T_i$  is the timestamp of the signature stage.  $V_i$  selects  $b_i \in Z_q^*$  randomly and computes  $B_i = b_i P$ ,  $M_p = M_{V_i} P$ , and  $A_i = N_{R_j} psk_i + (x_i + b_i) H_4^i$ . Then, the signature of the message  $M_i$  is  $\sigma_i = (A_i, B_i, M_p)$ . Vehicle  $V_i$  sends  $\{M_i, \sigma_i, PID_i, T_i\}$  to  $RSU_j$
- (2) RSUs deploy a smart contract to verify the correctness and integrity of signature message data
- (3) Blockchain nodes execute the algorithm. At first, the  $RSU_j$  determines the freshness of time stamp  $T_i$ . If  $T_i$  is not fresh, the message is discarded, and the operation is stopped. Otherwise,  $RSU_j$  computes  $H_4^{i'} = h_4(M_{V_i}^{-1} N_{R_j} M_{V_i} P || M_i || PID_i || T_i)$ . If  $H_4^{i'} \neq H_4^i$ ,

```

Input:  $\{M_i, \sigma_i, \text{PID}_i, T_i\}$ 
Output: True or False
1: begin
2:   if  $T_i$  is not fresh then
3:     throw;
4:   if  $H_4^i \neq H_4^i$  then
5:     throw;
6:   if  $A_i P - B_i H_4^i = N_{R_j}(K_{\text{pub}} + \text{PID}_i T_{\text{pub}}) + V_{\text{pub}_i} H_4^i$  then
7:     return True;
8:   end if

```

ALGORITHM 1: Smart contract algorithm in verification phase.

then it determines whether (4) is valid. If so, RSU<sub>j</sub> accepts  $\{M_i, \sigma_i, \text{PID}_i, T_i\}$ :

$$A_i P - B_i H_4^i = N_{R_j}(K_{\text{pub}} + \text{PID}_i T_{\text{pub}}) + V_{\text{pub}_i} H_4^i. \quad (4)$$

The following is the proof of correctness:

$$\begin{aligned}
A_i P - B_i H_4^i &= N_{R_j} \text{psk}_i P + (x_i + b_i) H_4^i P - B_i H_4^i \\
&= N_{R_j}(u + \text{PID}_i s) P + x_i H_4^i P + b_i H_4^i P - B_i H_4^i \\
&= N_{R_j}(K_{\text{pub}} + \text{PID}_i T_{\text{pub}}) + V_{\text{pub}_i} H_4^i + B_i H_4^i \\
&\quad - B_i H_4^i = N_{R_j}(K_{\text{pub}} + \text{PID}_i T_{\text{pub}}) + V_{\text{pub}_i} H_4^i
\end{aligned} \quad (5)$$

(4) The smart contract validates the data through a function interface and returns the correct result True only when  $H_4^i = H_4^i$  and  $A_i P - B_i H_4^i = N_{R_j}(K_{\text{pub}} + \text{PID}_i T_{\text{pub}}) + V_{\text{pub}_i} H_4^i$ . Otherwise, the error result False is returned. The smart contract verification algorithm in signature message verification phase can be written in Solidity language, and the content of the intelligent contract verification algorithm is roughly as shown in Algorithm 1

(5) If the verification is successful, RSU<sub>j</sub> packages the transaction  $Tx_2 = (H_4^i \| M_i \| \text{PID}_i \| T_i)$ , divides the transaction into several parts and stores them in different CS, and then performs  $h_5$  operation on  $Tx_2$  and stores it in the blockchain as CMT. Finally, the transaction is broadcasted to each node

**4.7. Tracking.** When traffic conditions are safe, but it is inconsistent with the transaction information, other vehicles can report the vehicles who published the false message. The RSUs then report the pseudoidentity  $\text{PID}_i = tH_1^i \bmod q$  of the malicious vehicle to TA. TA traces the real identity  $\text{RID}_i$  from his database by computing  $H_1^i = h_1(\text{RID}_i \| T_{\text{reg}})$  and revokes the user information  $(H_1^i, \text{PID}_i, \text{PID}_i s, T_{\text{reg}})$  to

update the user list UL accordingly. Finally, the RSU broadcasts the malicious vehicle and new UL to the blockchain.

## 5. Security Proof and Analysis

In this part, the security of our scheme under random oracle is proved, and we demonstrate that our basic scheme meets the security objectives in Section 3. In other words, our architecture can provide integrity, anonymity, unlinkability, and so on.

### 5.1. Security Proof

**Definition 3.** If adversaries A1 and A2 cannot win the following games with negligible probability in polynomial time, then the proposed scheme satisfies the Existential Unforgeability Against Adaptive Chosen Message Attack (EUF-CMA) in the random oracle model.

**Theorem 4.** If the running time of an adversary A1 with probability polynomial time in game 1 is  $t$ , execute  $Q_i (i = 1 \sim 4)$  hash queries,  $Q_{\text{PPK}}$  partial private key extraction queries, and  $Q_{\text{PK}}$  public key queries, and the advantage of forging a legal signature after  $Q_\sigma$  signature queries is  $\epsilon$ ; then, the ECDLP problem can be solved with a probability of no less than  $\epsilon/Q_{\text{PPK}}(1 - 1/Q_{\text{PPK}})^{Q_{\text{PPK}}}$  in  $t' \leq t + O(Q_2 + Q_3 + (Q_1 + Q_4 + Q_{\text{PPK}} + Q_{\text{PK}} + Q_\sigma)t_s)$  time, and  $t_s$  represents the time of one multiplication in group  $G$ .

**Proof of Theorem.** Challenger algorithm C first interacts with adversary A1 to generate an instance of the ECDLP problem, given  $P, Q = aP$ , where  $a \in Z_q^*, P \in G$ . The goal of challenger C is to solve for  $a$ . C interacts with A1, responds to all the queries of A1, and records these in the corresponding lists which are initially empty.

(i) Setup: challenger C initializes system parameter  $\{G, P, T_{\text{pub}}, K_{\text{pub}} = Q, h_1, h_2, h_3, h_4, h_5\}$  and sends the system parameter to A1 and C randomly selects  $ID^*$  as the challenge identity of the game. A1 makes the following inquiries:

- (1)  $h_1$  oracle: when A1 with an  $ID_i$  performs this query to  $h_1$  oracle, C records the questions and answers between A1 and C through list  $L_1 = (ID_i, h_1(ID_i, t))$ . If C looks up the corresponding  $L_1 = (ID_i, h_1(ID_i, t))$  in  $L_1$ , C returns  $h_1(ID_i, t)$  to A1; otherwise, C randomly picks  $h_1(ID_i, t') \in Z_q^*$  and sends it to A1 and then adds  $(ID_i, h_1(ID_i, t'))$  to the  $L_1$
- (2)  $h_2$  oracle: when A1 with a  $\text{PID}_i$  performs this query to  $h_2$  oracle, C records the questions and answers between A1 and C through list  $L_2 = (\text{PID}_i, \text{psk}_i^{-1} P, u_i)$ . If C looks up the corresponding  $(\text{PID}_i, \text{psk}_i^{-1} P, u_i)$  in  $L_2$ , C returns  $u_i$  to A1; otherwise, C randomly picks  $u_i \in Z_q^*$  and sends

- it to  $A_1$  and then adds  $(PID_i, \text{psk}_i^{-1}P, u_i)$  to the  $L_2$
- (3)  $h_3$  oracle: when  $A_1$  with  $(N_{R_j} \| M_{V_i})$  performs this query to  $h_3$  oracle,  $C$  records the questions and answers between  $A_1$  and  $C$  through list  $L_3 = (N_{R_j} \| M_{V_i}, v_i)$ . If  $C$  looks up the corresponding  $(N_{R_j} \| M_{V_i}, v_i)$  in  $L_3$ ,  $C$  returns  $v_i$  to  $A_1$ ; otherwise,  $C$  randomly picks  $v_i \in Z_q^*$  and sends it to  $A_1$  and then adds  $(PID_i, \text{psk}_i^{-1}P, u_i)$  to the  $L_3$
- (4)  $h_4$  oracle: when  $A_1$  with  $(N_P \| M_i \| PID_i \| T_i)$  performs this query to  $h_4$  oracle,  $C$  records the questions and answers between  $A_1$  and  $C$  through list  $L_4 = (N_P \| M_i \| PID_i \| T_i, w_i)$ . If  $C$  looks up the corresponding  $(N_P \| M_i \| PID_i \| T_i, w_i)$  in  $L_4$ ,  $C$  returns  $w_i$  to  $A_1$ ; otherwise,  $C$  randomly picks  $w_i \in Z_q^*$  and sends it to  $A_1$  and then adds  $(N_P \| M_i \| PID_i \| T_i, w_i)$  to the  $L_4$
- (5) Partial-private key oracle: when  $A_1$  with  $PID_i$  performs this query to partial-private key oracle,  $C$  records the questions and answers between  $A_1$  and  $C$  through list  $L_{\text{par}} = (PID_i, \text{psk}_i)$ . If  $C$  looks up the corresponding  $(PID_i, \text{psk}_i)$  in  $L_{\text{par}}$ ,  $C$  returns  $\text{psk}_i$  to  $A_1$ ; otherwise, if  $PID_i \neq PID_i^*$ ,  $C$  randomly picks  $\text{psk}_i \in Z_q^*$  and sends  $\text{psk}_i$  to  $A_1$  and then saves  $(PID_i, \text{psk}_i)$  in list  $L_{\text{par}}$ . If  $PID_i = PID_i^*$ ,  $C$  stops the game
- (6) Public key oracle: when  $A_1$  with  $PID_i$  performs this query to partial-private key oracle,  $C$  records the questions and answers between  $A_1$  and  $C$  through list  $L_{\text{pub}} = (PID_i, x_i, V_{\text{pub}_i})$ . If  $C$  looks up the corresponding  $(PID_i, x_i, V_{\text{pub}_i})$  in  $L_{\text{pub}}$ ,  $C$  returns  $V_{\text{pub}_i}$  to  $A_1$ ; otherwise, if  $PID_i \neq PID_i^*$ ,  $C$  randomly picks  $x_i \in Z_q^*$ , let  $V_{\text{pub}_i} = x_i P$  and sends  $V_{\text{pub}_i}$  to  $A_1$ , and then saves  $(PID_i, x_i, V_{\text{pub}_i})$  in list  $L_{\text{pub}}$
- (7) Secret value oracle: when  $A_1$  with  $PID_i$  performs this query to secret value oracle, if  $PID_i \neq PID_i^*$ ,  $C$  gives up and terminates the operation; otherwise,  $C$  looks for list  $L_{\text{pub}}$ , and if record  $(PID_i, x_i, V_{\text{pub}_i})$  exists,  $C$  returns  $x_i$  to  $A_1$ . If not,  $C$  performs a public key query to generate tuple  $(x_i, V_{\text{pub}_i})$ , returns  $x_i$  to  $A_1$ , and adds  $(x_i, V_{\text{pub}_i})$  to the  $L_{\text{pub}}$
- (8) Replace public key oracle: when  $A_1$  with  $(PID_i, V_{\text{pub}_i}')$  performs this query to replace public key oracle,  $C$  first finds the corresponding record  $(PID_i, x_i, V_{\text{pub}_i})$  from  $L_{\text{pub}}$ . If it does not exist,  $C$  performs public key query to generate tuple  $(x_i, V_{\text{pub}_i})$ , returns  $x_i$  to  $A_1$ , and adds  $(x_i, V_{\text{pub}_i})$  to the  $L_{\text{pub}}$
- (9) Signature oracle:  $A_1$  performs signature oracle with  $(PID_i, M_i)$ , and  $C$  recovers  $h_1(ID_i, T_{\text{reg}})$ ,  $h_2(PID_i \| \text{psk}_i^{-1}P)$ , and  $h_3(N_{R_j} \| M_{V_i}), h_4(N_P \| M_i \| PID_i \| T_i)$  from list  $L_1, L_2, L_3, L_{\text{par}}, L_{\text{pub}}$ ; if  $PID_i \neq PID_i^*$ , then  $C$  outputs the signature  $\sigma_i$  corresponding to message  $M_i$  and transmits  $\sigma_i$  to  $A_1$ ; otherwise,  $C$  randomly picks  $b_i \in Z_q^*$  and computes  $B_i = b_i P$ ,  $M_P = M_{V_i} P$ , and  $A_i = N_{R_j} \text{psk}_i + (x_i + b_i)H_4^i$ .  $\sigma_i = (A_i, B_i, M_P)$  represents a correct signature of the signer to the message  $M_i$ . And then finally,  $C$  returns  $\sigma_i$  to  $A_1$
- (ii) Forgery: finally,  $A_1$  prints a forged signature. If  $PID_i \neq PID_i^*$ ,  $C$  stops the simulation; otherwise,  $C$  finds the corresponding signature information  $\{M_i, \sigma_i, PID_i, T_i\}$  from the prophecy query list, and if adversary  $A_1$  wins the game, then  $A_i P - B_i H_4^i = N_{R_j} (K_{\text{pub}} + PID_i T_{\text{pub}}) + V_{\text{pub}_i} H_4^i$ . Then, the bifurcation lemma [26] is used to obtain another two sets of valid signatures  $\sigma_i^{(\ell)}$ ,  $(\ell = 2, 3)$  in polynomial time, and these three signatures must satisfy  $A_i P - B_i H_4^i = N_{R_j} (K_{\text{pub}} + PID_i T_{\text{pub}}) + V_{\text{pub}_i} H_4^i$ . And because of  $V_{\text{pub}_i} = x_i P$ ,  $B_i = b_i P$ , and  $K_{\text{pub}} = aP$ , there are three linearly independent equations:
- $$\begin{aligned} A_i^\ell \cdot P - B_i^\ell H_4^i &= N_{R_j} (K_{\text{pub}} + PID_i^\ell T_{\text{pub}}) \\ &+ V_{\text{pub}_i}^\ell H_4^i, \ell = 1, 2, 3 \end{aligned} \quad (6)$$
- Challenger  $C$  solves the solution of these three equations and takes the  $a$  output as the solution of the ECDLP problem.
- In the partial private key extraction query, the probability of  $C$  not giving up is at least  $(1 - 1/Q_{\text{PPK}})^{Q_{\text{PPK}}}$ , and the probability of  $C$  not giving up in the forgery stage is at least  $1/Q_{\text{PPK}}$ . Therefore, the probability of  $C$  successfully solving the problem is at least  $\varepsilon/Q_{\text{PPK}}(1 - 1/Q_{\text{PPK}})^{Q_{\text{PPK}}}$ . The running time of  $C$  is  $t' \leq t + O(Q_2 + Q_3 + (Q_1 + Q_4 + Q_{\text{PPK}} + Q_{\text{PK}} + Q_\sigma)t_s)$ .  $\square$
- Theorem 5.** *If the running time of an adversary  $A_2$  with probability polynomial time in game 2 is  $t$ , execute  $Q_i (i = 1 \sim 4)$  hash queries,  $Q_X$  secret value queries, and  $Q_{\text{PK}}$  public key queries, and the advantage of forging a legal signature after  $Q_\sigma$  signature queries is  $\varepsilon$ ; then, the ECDLP problem can be solved with a probability of no less than  $\varepsilon/Q_{\text{PK}}(1 - 1/Q_{\text{PK}})^{Q_{\text{PK}} + Q_X}$  in  $t' \leq t + O(Q_2 + (Q_1 + Q_3 + Q_{\text{PK}} + Q_\sigma)t_s)$  time, and  $t_s$  represents the time of one multiplication in group  $G$ .*
- The idea and method of proof are similar as that of Theorem 4.  $A_2$  only has the ability of hash value inquiry, public key extraction inquiry, secret value inquiry, private key extraction inquiry, and signature inquiry but does not have the

ability of public key substitution, so the proof process will not be repeated here.

## 5.2. Security Analysis

**5.2.1. Message Authentication.** The RSU or any vehicle within the RSU domain can verify the message  $M_i$  by verifying the pseudoidentity  $PID_i$  of the vehicle and using its signature  $\sigma_i$ . In addition, under the condition that ECDLP is difficult, there is no polynomial-time adversary to forge valid signatures; that is, there is no probabilistic polynomial-time adversary to forge valid messages without using the private key of signature. Therefore, the receiver can verify the authenticity and integrity of message  $\{M_i, \sigma_i, PID_i, T_i\}$  by verifying whether  $H_4^i$  and  $H_4^i$  are equal to each other and whether (1) holds.

**5.2.2. Anonymity.** The vehicle  $V_i$  in the IoV environment with pseudoidentity  $PID_i$  sends a message  $\{M_i, \sigma_i, PID_i, T_i\}$ , where  $PID_i = tH_1^i \bmod q$ ,  $H_1^i = h_1(RID_i || T_{reg})$ , and  $T_{reg}$  are the registration time of the vehicle, and  $t$  is randomly selected from  $Z_q^*$ . To extract the real identity  $RID_i$ , the adversary should calculate  $H_1^i$ , and  $t$  is random. Obviously, it is impossible for the enemy to calculate the  $RID_i$  of vehicle  $V_i$  for discrete logarithm problem and one-way hash function. Therefore, the scheme guarantees the privacy of the vehicle.

**5.2.3. Unlinkability.** The message  $M$  and message  $M'$  are signed by different private keys  $SK^i$ , and the pseudoidentity  $PID_i$  is calculated by TA,  $PID_i = tH_1^i \bmod q$ , where  $H_1^i$  is the hash value of the real identity  $RID_i$  and the registration time  $T_{reg}$ , and  $t$  is a random number, so it is impossible for any adversary to link any pseudoidentity  $PID_i$ .

**5.2.4. Forward Security and Backward Security.** In this scheme, if the adversary obtains the signed message  $\sigma_i = (A_i, B_i, M_p)$ , where  $A_i = N_{R_j} \cdot psk_i + (x_i + b_i)H_4^i$ ,  $B_i = b_iP$ , and  $M_p = M_{V_i}P$ , because  $b_i$  is random, and  $N_{R_j}$  is a random integer secretly negotiated between the  $RSU_j$  and vehicle  $V_i$ . So each signature is different and the adversary cannot infer a previous or subsequent signature message from the current signature message.

**5.2.5. Traceability.** The proposed scheme provides conditional identity privacy preservation in IoVs. TA tracked and revealed the real identity of the malicious vehicle from its database. When a malicious vehicle with forged messages is found, the real identity of the malicious vehicle can be traced by TA. TA, as a trust authority, cannot tamper the real identity and the pseudoidentity  $PID_i$  of a malicious vehicle because tuple  $(H_1^i, PID_i, PID_i, T_{reg})$  of the vehicle is recorded in a blockchain.

**5.2.6. Cross-Domain Authentication.** First, when the vehicle  $V_i$  publishes a message  $M_i$ , it sends the signature  $\sigma_i = (A_i, B_i, M_p)$  to the first RSU of  $M_i$ . If the verification is successful, a new transaction  $Tx_2 = (H_4^i || M_i || PID_i || T_i)$  is generated and  $h_5(Tx_2)$  is recorded on the blockchain and broadcasted to all the nodes. When other RSUs and vehicles in other RSUs

receive the message  $M_i$ , they only need to look up the verification information related to the message from the chain. Second, only the first RSU negotiates a random integer  $N_{R_j}$  with a vehicle  $V_i$ , and a transaction  $Tx_1 = (PID_i || V_{pub_i} || L_{ij})$  with  $L_{ij} = M_{V_i}^{-1}N_{R_j}$ ,  $PID_i$ , and  $V_{pub_i}$  is recorded on the blockchain. Therefore, when the vehicle  $V_i$  enters the next RSU range, if a new message is to be published, there is no need to renegotiate the random number.

## 5.2.7. Resilience to Other Attacks

(1) *Impersonation Attack.* In order to impersonate a registered vehicle, the enemy needs a valid signature for message  $M_i$ . Therefore,  $N_p = N_{R_j}P$ ,  $H_4^i = h_4(N_p || M_i || PID_i || T_i)$ ,  $B_i = b_iP$ ,  $M_p = M_{V_i}P$ , and  $A_i = N_{R_j} \cdot psk_i + (x_i + b_i)H_4^i$  need to be calculated, where  $N_{R_j}$  is an integer to ensure the integrity of the transaction content negotiated between vehicle  $V_i$  and  $RSU_j$ , and the adversary needs to know the private key used for signature. Therefore, in calculation, it is not feasible for the adversary to create another valid request message without knowing the above content. Therefore, the scheme is safe from vehicle simulation attacks.

(2) *Modification Attack.* In the authentication phase, the freshness of the message is determined by the timestamp at the time of signing, and the message  $\{M_i, \sigma_i, PID_i, T_i\}$  sent by vehicle  $V_i$  has the timestamp  $T_i$  of the sender. Any nearby vehicle or RSU can check  $T_i$  to verify the freshness of the message. It prevents the message  $\{M_i, \sigma_i, PID_i, T_i\}$  from being repeatedly broadcast in the RSU domain, so the scheme protects against replay attacks.

(3) *Man-In-The-Middle Attack.* Suppose an adversary intercepts a message  $\{M_i, \sigma_i, PID_i, T_i\}$ , and the adversary tries to create a valid signature in place of the vehicle to send to the RSU as an authentication request. However, it can be seen from the signature that the adversary needs to know  $N_{R_j}$ ,  $b_i$ , and  $M_{V_i}$ , and these parameters are embedded in the ECDLP difficulty problem, so the scheme is not subject to man-in-the-middle attack.

(4) *Replay Attack.* For the signature  $\sigma_i = (A_i, B_i, M_p)$  of the message  $M_i$  and  $A_i = N_{R_j} \cdot psk_i + (x_i + b_i)H_4^i$  and  $H_4^i = h_4(N_p || M_i || PID_i || T_i)$ , the message contains the current timestamp; when receiving a message, the RSU checks the validity of the message by comparing the received timestamp to the current timestamp. For valid message and its freshness, the difference between the timestamp should be a small value; therefore, by containing a timestamp in each message, one can ensure that the message is protected from replay attacks because no adversaries can successfully replay the intercepted message.

## 6. Performance Analysis

This section will give a function comparison, then analyze and calculate the communication costs of our scheme and

TABLE 2: Function comparison.

Scheme	Privacy	Decentralization	Multistorage	Anonymity	Unforgeability	Key escrow
Zheng et al. [3]	Yes	Yes	Yes	Yes	-	Yes
Malhi and Batra [12]	Yes	-	-	Yes	Yes	No
Bagga et al. [19]	Yes	Yes	No	Yes	Yes	Yes
Ren et al. [20]	Yes	Yes	No	No	Yes	No
Ours	Yes	Yes	Yes	Yes	Yes	No

related schemes, and make experimental simulation and analysis.

In Table 2, we first give a simplified comparison of functions between our scheme and other related schemes [3, 12, 19, 20] in terms of privacy, decentralization, multistorage, anonymity, unforgeability, and key escrow. We use the symbol “-” to represent that the corresponding property is not considered. In scheme [3], it realises the privacy protection, decentralization, multistorage, and anonymity of vehicles in the authentication process but does not provide unforgeability, nor could it avoid the key escrow. And in scheme [12], it does not consider decentralization and multistorage. The privacy protection, decentralization, and anonymity of vehicles are implemented in the scheme [19]. However, multistorage and key escrow are not considered. As to scheme [20], it achieved the privacy protection, decentralization, unforgeability, and avoided key escrow but did not have multistorage and anonymity. As shown in Table 2, our scheme satisfies all the above functions.

**6.1. Computational Cost Analysis.** In terms of computational overhead, we analyze our scheme and compare it with the recent correlative signature schemes [3, 12, 19, 20] for signature generation and verification in V2I communication.

We use a MIRACL simulation library in VS 2019. The simulation environment is Win1064 bit, and the hardware environment is Intel Core i5 3.10 GHz.  $T_{bp}$  denotes the execution time of a bilinear pairing operation,  $T_{sm}$  denotes the execution time of a scalar multiplication operation in  $G_1$ ,  $T_{pa}$  denotes the execution time of a point addition operation in  $G_1$ ,  $T_{pm}$  denotes the execution time of a point multiplication operation in  $G_1$ , and hash operation is not considered for its low load. The addition and multiplication of points on an elliptic curve are performed under a nonsingular elliptic curve  $y^2 = x^3 + ux + v \pmod{q}$ , where  $4u^3 + 27v^2 \neq 0 \pmod{q}$ .

Table 3 shows the computation costs of the proposed scheme and schemes [12, 19, 20] in the signature and verification stage. As for scheme [3], it is defaulted that only legal vehicles can publish messages. Therefore, vehicles only submit messages without signing and RSU does not need to perform identity verification. Because only hash operation is included, the computation is very low, but this is exactly a security vulnerability. In [12], users need to perform four scalar multiplication operations and two point addition operations at the signing stage; that is, users need  $4T_{sm} + 2T_{pa}$  in total and RSUs need  $3T_{bp} + 3T_{sm} + T_{pa}$ . The computation cost is  $3T_{pm} + 3T_{pa}$ ,  $3T_{bp} + 5T_{pm} + T_{pa}$ ,  $2T_{sm}$ , and 2

TABLE 3: Computation cost comparison.

Scheme	Sign (ms)	Verify (ms)
Zheng et al. [3]	-	-
Malhi and Batra [12]	$4T_{sm} + 2T_{pa}$	$3T_{bp} + 3T_{sm} + T_{pa}$
Bagga et al. [19]	$3T_{pm} + 3T_{pa}$	$3T_{bp} + 5T_{pm} + T_{pa}$
Ren et al. [20]	$2T_{sm}$	$2T_{bp} + T_{pa}$
Ours	$3T_{pm} + 2T_{sm}$	$5T_{pm} + 2T_{pa}$

$T_{bp} + T_{pa}$  in [19, 20], respectively. Our proposed scheme needs three point multiplication operations on elliptic curves and one scalar multiplication (i.e.,  $3T_{pm} + 2T_{sm}$ ) in the signature process. In the verification process, the computational cost is  $5T_{pm} + 2T_{pa}$ . Our scheme is built on elliptic curves with less computation overhead. There is no bilinear pairing operation with higher computation overhead in the signature and verification phase. Only point and multiplication operations with lower computation overhead are used. Therefore, the computation overhead of this scheme is better than the other three schemes in the signature and verification phase.

**6.2. Communication Cost Analysis.** To analyze the communication overhead of the proposed scheme and the related signature schemes [12, 19, 20], we analyze the communication overhead by considering the size of parameters. At the security level of 80 bytes, the  $p$ -length equation  $E: y^2 \equiv (x^3 + x) \pmod{p}$  is 64 bytes, and the elements on the circle group  $G$  occupy 128 bytes. We consider the size of the timestamp to be 4 bytes and the size of the normal hash function to be 20 bytes. Table 4 shows the communication costs of the proposed scheme and schemes [3, 12, 19, 20]. As can be seen from Table 4, the communication overhead of Zheng et al. is the lowest in these five schemes because only message is transmitted without signature. Then, our scheme is significantly lower than that of Bagga et al. and Ren et al. Although the communication cost of the proposed scheme is the same as that of Malhi and Batra, the computation is significantly lower than that of Malhi and Batra. Therefore, our scheme has more communication advantages.

**6.3. Experimental Simulation and Analysis.** This section uses the MIRACL library to test the computation costs in Table 3 in Visual Studio 2019. Figures 8 and 9 show the relationship between the number of message and the time consumed by the proposed scheme and the other three schemes in the

TABLE 4: Communication cost comparison.

Scheme	Single-Msg.	n-Msg.
Zheng et al. [3]	316 bytes	$316n$ bytes
Malhi and Batra [12]	536 bytes	$536n$ bytes
Bagga et al. [19]	664 bytes	$664n$ bytes
Ren et al. [20]	660 bytes	$660n$ bytes
Ours	536 bytes	$536n$ bytes

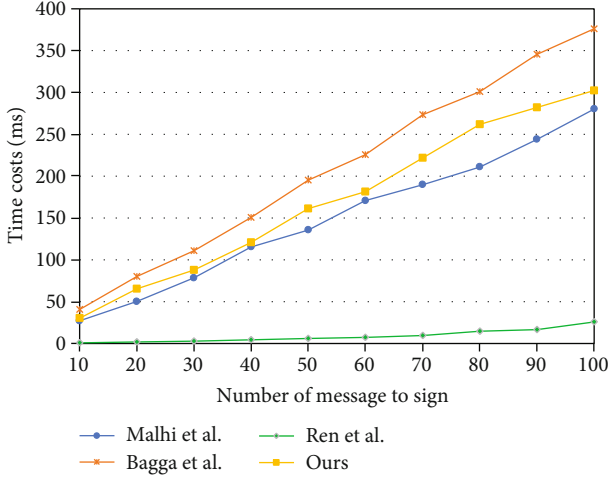


FIGURE 8: Signature phase computation costs.

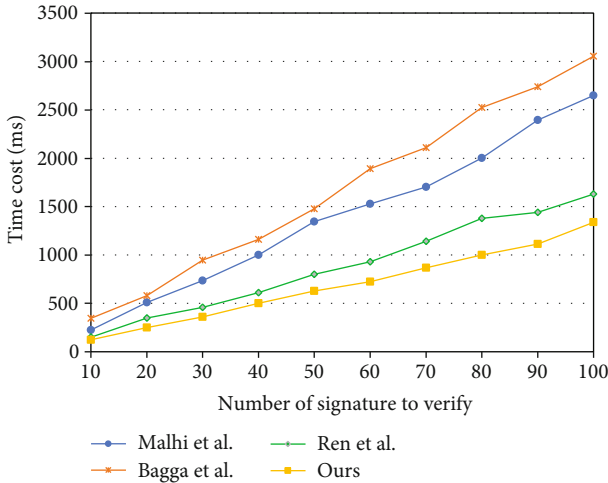


FIGURE 9: Validation phase computation costs.

signing and authentication process. It can be seen from Figures 8 and 9 that our scheme has no significant advantages in the signature phase but has obvious advantages in the verification phase, because in our scheme, only the message needs to be authenticated, rather than the vehicle identity needs to be authenticated. Specifically, the vehicle identity information is authenticated by the first RSU, and transactions are recorded permanently and immutably in the blockchain to reduce the repeated authentication load of other RSUs.

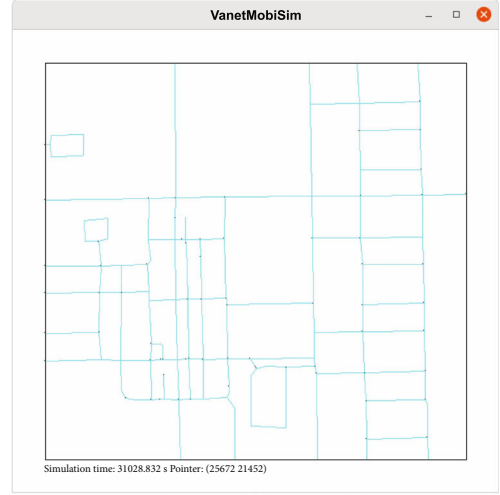


FIGURE 10: Scene simulation diagram.

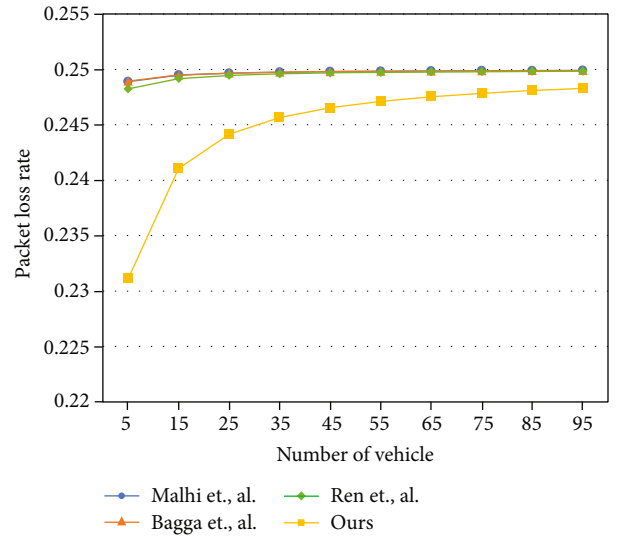


FIGURE 11: Packet loss rate comparison.

Through theoretical analysis and simulation experiments, it is intuitive to see that with the increase of messages, the proposed scheme has certain efficiency advantages over the schemes [12, 19, 20].

The packet loss rate and delay are tested according to the time and traffic of the scheme in the signature and verification stage. A scenario is simulated by using VanetMobiSim and NS-2, as shown in Figure 10. The scenario is divided into four parts, and each part is managed by an RSU. The vehicle speed is controlled between 7 m/s and 45 m/s, the communication range is 400 m, the message interval is 80 ms, and the data packet sizes are 536 bytes, 664 bytes, 660 bytes, and 536 bytes, respectively.

When testing the packet loss rate and time delay, set the number of vehicles to gradually increase from 5 to 95. The simulation results of packet loss rate are shown in Figure 11. The packet loss rate in the four schemes is also gradually increasing as the increase of the number of

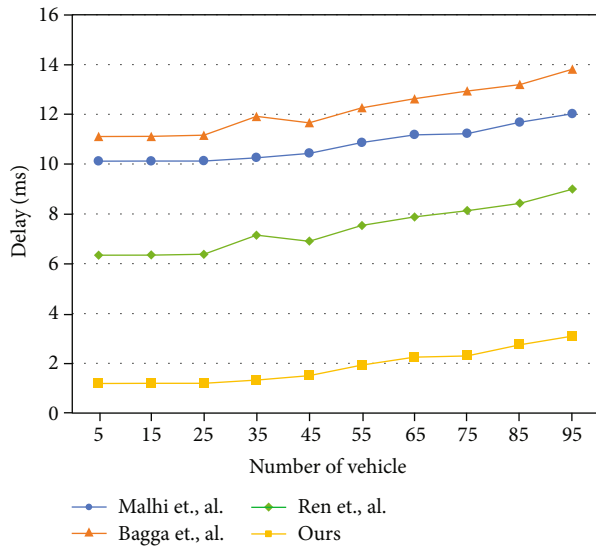


FIGURE 12: Delay comparison.

vehicles. When the quantity of vehicles reaches a certain number, the packet loss rate tends to be stable. Compared with the other three schemes, our scheme has the lowest packet loss rate, which shows that our scheme has high efficiency. The delay results are shown in Figure 12. Although the delay of the four schemes increases gradually, the delay of our scheme is the smallest. Therefore, the communication timeliness of our scheme is higher than that of the other three schemes.

To sum up, through theoretical analysis and simulation experiments, our scheme has certain advantages in traffic, computation, packet loss rate, and delay, so it is suitable for the IoVs with high security and high efficiency requirements.

## 7. Conclusion and Future Research

On the basis of the scheme [3], we put forward an effective cross-domain certificateless anonymous authentication scheme based on blockchain by using pairing-free signature verification scheme. Our scheme reduces the computing cost of signature verification on RSUs, solves the key escrow problem in traditional authentication scheme, and improves the efficiency of V2I communication. In addition, the blockchain built by RSUs is introduced to the scheme, the vehicle identity and messages realise cross-domain authentication through a random integer negotiation process, and the blockchain stores the identity information of the vehicle and the authentication results of the signature. As a result, the load and delay caused by repeated identity and message authentication are prevented. Our scheme is provably secure and provides integrity, anonymity, privacy, traceability, revocation, and nonrepudiation. Experiments show that our scheme is efficient in terms of computation costs, latency, and packet loss rate for signature generation and verification.

Compared with cloud computing, edge computing is very faster, reduces network latency, and is more reliable.

Therefore, the future works are suggested to study anonymous authentication based on edge computing in IoVs.

## Data Availability

All relevant data are within the paper.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 61662071 and 61772022).

## References

- [1] "Dedicated, short range communications (DSRC)," <http://grouper.ieee.org/groups/scc32/dsrc/index.html>.
- [2] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology - ASIACRYPT 2003. ASIACRYPT 2003. Lecture Notes in Computer Science*, C. S. Lai, Ed., vol. 2894, Springer, Berlin, Heidelberg, 2003.
- [3] D. Zheng, C. Jing, R. Guo, S. Gao, and L. Wang, "A traceable blockchain-based access authentication system with privacy preservation in VANETs," *IEEE Access*, vol. 7, pp. 117716–117726, 2019.
- [4] A. Wasef and X. Shen, "EMAP: expedite message authentication protocol for vehicular ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 1, pp. 78–89, 2013.
- [5] S. Wang, N. Yao, N. Gong, and Z. Gao, "A trigger-based pseudonym exchange scheme for location privacy preserving in VANETs," *Peer-to-Peer Networking and Applications*, vol. 11, no. 3, pp. 548–560, 2018.
- [6] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology. CRYPTO 1984. Lecture Notes in Computer Science*, G. R. Blakley and D. Chaum, Eds., vol. 196, Springer, Berlin, Heidelberg, 1985.
- [7] M. Bayat, M. Barmshoory, M. Rahimi, and M. R. Aref, "A secure authentication scheme for VANETs with batch verification," *Wireless Networks*, vol. 21, no. 5, pp. 1733–1743, 2015.
- [8] C. Song, M. Zhang, W. Penget, Z. Jia, and X. Yan, "Research on batch anonymous authentication scheme for VANET based on bilinear pair," *Journal on Communications*, vol. 38, no. 6, pp. 49–57, 2017.
- [9] S. Tzeng, S. Horng, T. Li, X. Wang, P. Huang, and M. K. Khan, "Enhancing security and privacy for identity-based batch verification scheme in VANETs," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3235–3248, 2017.
- [10] Y. Yao, X. Chang, L. Li, and R. Yang, "CLAM: lightweight certificateless anonymous authentication mechanism for vehicular cloud services," *Cyber-Physical Systems*, vol. 4, no. 1, pp. 17–38, 2018.
- [11] G. Xu, W. Zhou, A. K. Sangaiah et al., "A security-enhanced certificateless aggregate signature authentication protocol for InVANETs," *IEEE Network*, vol. 34, no. 2, pp. 22–29, 2020.
- [12] A. K. Malhi and S. Batra, "An efficient certificateless aggregate signature scheme for vehicular ad-hoc networks," *Discrete Mathematics and Theoretical Computer Science*, vol. 17, no. 1, pp. 317–338, 2015.

- [13] D. Wang and J. Teng, "Probably secure certificateless aggregate signature algorithm for VANET," *Journal of Electronics and Information Technology*, vol. 40, no. 1, pp. 11–17, 2018.
- [14] X. Yang, T. Ma, C. Chen, J. Wang, and C. Wang, "Security analysis and improvement of certificateless aggregate signature scheme for vehicular adhoc network," *Journal of Electronics and Information Technology*, vol. 41, no. 5, pp. 1265–1270, 2019.
- [15] N. Zhao and G. Zhang, "Authenticated privacy protection scheme based on certificateless ring signcryption in VANET," *Computer Science*, vol. 47, no. 3, pp. 312–319, 2020.
- [16] W. HATHAL, H. Cruickshank, Z. Sun, and C. Maple, "A Certificateless and lightweight authentication scheme for vehicular communication networks," *IEEE Transactions on Vehicular Technology*, vol. 99, pp. 1–16, 2020.
- [17] Y. Yang, J. Zhang, and J. Ma, "A method of using blockchain to protect data privacy of Internet of vehicle," *Journal of Xidian University*, vol. 48, no. 3, pp. 21–30, 2021.
- [18] I. Ali, M. Gervais, E. Ahene, and F. Li, "A blockchain-based certificateless public key signature scheme for vehicle-to-infrastructure communication in VANETs," *Journal of Systems Architecture*, vol. 99, article 101636, 2019.
- [19] P. Bagga, A. K. Sutrala, A. K. Das, and P. Vijayakumar, "Blockchain-based batch authentication protocol for Internet of vehicles," *Journal of Systems Architecture*, vol. 113, no. 8, article 101877, 2020.
- [20] Y. Ren, X. Li, S. Sun, X. Yuan, and X. Zhang, "Privacy-preserving batch verification signature scheme based on blockchain for vehicular ad-hoc networks," *Journal of Information Security and Applications*, vol. 58, article 102698, 2021.
- [21] P. Vijayakumar, M. Azees, S. A. Kozlov, and J. J. P. C. Rodrigues, "An anonymous batch authentication and key exchange protocols for 6G enabled VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1630–1638, 2022.
- [22] M. Azees, A. S. Rajasekaran, and M. I. Satti, "An anonymous signature-based authentication and key agreement scheme for vehicular ad hoc networks," *Security and Communication Networks*, vol. 2022, Article ID 1222660, 9 pages, 2022.
- [23] S. M. Pournaghi, B. Zahednejad, M. Bayat, and Y. Farjami, "NECPPA: a novel and efficient conditional privacy-preserving authentication scheme for VANET," *Computer Networks*, vol. 134, pp. 78–92, 2018.
- [24] B. C. Hu, D. S. Wong, Z. Zhang, and X. Deng, "Key replacement attack against a generic construction of certificateless signature," in *Information Security and Privacy. ACISP 2006. Lecture Notes in Computer Science*, L. M. Batten and R. Safavi-Naini, Eds., vol. 4058, Springer, Berlin, Heidelberg, 2006.
- [25] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, Consulted, 2008.
- [26] P. David and S. Jacques, "Security arguments for digital signatures and blind signatures," *Journal of Cryptology*, vol. 13, no. 3, pp. 361–369, 2000.