Peer-to-Peer Networking and Applications (2023) 16:2349–2366
https://doi.org/10.1007/s12083-023-01532-7

Traffic data security sharing scheme based on blockchain and traceable ring signature for VANETs

Xiaohong Zhang[1] · Jiaming Lai[1] · Ata Jahangir Moshayedi[2]

Received: 30 December 2022 / Accepted: 14 July 2023 / Published online: 5 August 2023
© The Author(s) 2023

Abstract

Vehicular ad hoc networks (VANETs) is the hotspot research field of wireless mobile ad hoc network, it provides a new opportunity to create a safe and efficient transportation environment. However, as an open network where information has to interact frequently, it is difficult to ensure the security of data transmitted in VANETs and protect the privacy of drivers. Many existing information-sharing schemes use complex encryption algorithms to enable secure traffic data sharing. Nevertheless, these schemes are not suitable for VANETs because of their high computational overhead and lack of corresponding tracking mechanisms for malicious vehicles. Therefore, a traffic data security sharing scheme is designed that combines blockchain technology and traceable ring signature algorithms to secure the transmitted messages. The traceable ring signature algorithm is formulated in combination with bilinear pairing, enabling conditional privacy protection instead of traditional ring signature. To improve the efficiency of VANETs, this scheme introduces edge computing technology to reduce the computational burden of Road Side Units (RSUs) by offloading most of the computational tasks to the servers via edge nodes. In addition, we use smart contract to track malicious vehicles. Security analysis and performance comparison show that our scheme is more efficient and secure for drivers than other existing related schemes.

Keywords Blockchain · Traceable ring signature · Edge computing · Smart contract · VANETs

1 Introduction

Intelligent transportation system [1] integrates electronic, communication and computer technologies to build a real-time, accurate and efficient traffic operation frame. Vehicular ad hoc networks (VANETs) describe the mobility and coexistence of each node vehicle-to-vehicle and vehicle-to-roadside communication architectures, and provide self-organized data transmission, safety precaution, navigation and other roadside services [2]. Vehicles exchange data through short-range wireless communication, this real-time information interaction (such as traffic information, weather conditions and road status, etc.) can help vehicles or traffic control centers to take on-line action to reduce traffic accidents or road congestion. Therefore, more and more scholars are devoted to the research of VANETs to improve people's quality of life and work efficiency.

In VANETs, all vehicles are equipped with On-Board Units (OBUs) and Tamper-Proof Devices (TPDs) [3, 4]. The OBUs are mainly organized by components such as resource processors, storage units, sensors and correspondence interface moduls, which are responsible for information exchange with surrounding vehicles and roadside infrastructure. The TPDs are used to store the encrypted information of the vehicle and prevent attackers from maliciously tampering with raw information or confidential data. With the continuous emergence of the Internet of Things, 5G, big data, cloud computing and other high-technolog, the communication performance of VANETs has been raised to a higher level. Vehicles equipped with OBU can realize vehicle-to-everything (V2X) mobile communication, which mainly includes vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) information interaction. They adopt

This article is part of the Topical Collection: *3 - Track on Blockchain*
Guest Editor: Haojin Zhu

✉ Xiaohong Zhang
xiaohongzh@jxust.edu.cn

Jiaming Lai
2422895407@qq.com

Ata Jahangir Moshayedi
moshayedi@iaukhsh.ac.ir

[1] School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou 341000, China

[2] KhomeiniShahr Branch, Islamic Azad University, Isfahan 10587, Iran

Dedicated Short-Range Communication (DSRC) protocol [5], and usually have a communication range of less than 1000 m.

In order to enhance the vehicle's ability to dynamically update its driving status, it often functions as a network node that gathers relevant information and utilizes wireless communication techniques to transmit this data to other nearby vehicles. This way enables the vehicle to proactively avoid potential accidents and mitigate traffic congestion, thereby improving overall driving efficiency. In addition, the vehicle will also send the collected information to Road Side Units (RSUs), which will summarize its own basic data and then upload them to the traffic control center (TCC). TCC is in charge of processing the data and then controlling the road signals or broadcasting vehicles' status to improve the efficiency of traffic. However, due to the open nature of VANETs, vehicle nodes share data in insecure channels, which can be intercepted, relayed or even tampered the transmitted data by malicious entities [6], the result is information suspicious or traffic chaos. And due to the limited computing and storage resources of the OBU, the rapid movement of the vehicle leads to real-time changes in the network topology, resulting in untimely message processing and undesirable consequences.

Therefore, secure authentication of messages transmitted and reduced system latency in VANETs is a very critical requirement, Chen and Chen [7] offer a private information visit only from trusted authority (TA), even though the traffic management department (TMD) wants to access or obtain the private messages. The convergence of blockchain and edge computing paradigms [8] can overcome the existing security and scalability issues. The scheme proposed by Ayaz et al. [9] uses edge nodes as mining nodes to solve the hash problem in the blockchain generation process. In the past few years, many scholars have proposed privacy-preserving anonymous authentication protocols [10–12]. However, we observe that the previously proposed protocols are highly dependent on a centralized server.

Blockchain technology [13, 14] derived from Bitcoin is a distributed database jointly maintained by multiple parties, with advantages such as decentralization, immutability, and traceability compared to traditional databases. Edge computing (EC) [15] is envisioned as a promising paradigm for processing massive amounts of data generated by ubiquitous mobile devices to enable intelligent services with artificial intelligence (AI). Multi-Access Edge Computing (MEC) [16, 17] is emerging as a key technology to provide low latency, high speed and high capacity network services for VANETs. The foundation of VANETs is information sharing [18], but due to the lack of trust between vehicles and the insecurity of the communication environment, blockchain technology has emerged to provide a viable scheme for these problems. Due to the nature of blockchain technology, drivers are able to remain anonymous in order to protect their privacy, but this prevents messages transmitted in VANETs from being verified, so digital signature [19] are a great way to verify messages.

Based on the above analysis, we propose a traffic data secure sharing scheme based on blockchain and traceable ring signature for VANETs. The main contributions of this work are as follows:

1. We propose a traceable ring signature algorithm by combining bilinear pairing and a ring signature algorithm. The algorithm uses the idea of distributed key generation to generate the user's key. In addition, the signature algorithm achieves conditional anonymity, this is to prevent disruption of normal traffic orders by malicious vehicles in VANETs.
2. We design a traffic data security sharing scheme. The scheme combines blockchain and the proposed traceable ring signature algorithm to enable secure sharing of traffic data with source traceability and conditional privacy protection.
3. Edge computing incorporated into our scheme. Through edge nodes, most computing and storage tasks are outsourced to cloud servers. It can effectively deal with the insufficient computing and storage capacity of its OBUs by using them as light nodes. The cloud is able to reduce the latency of VANETs and improve traffic efficiency since it has more computing power and storage capacity.
4. We use smart contract to track malicious users. We write the proposed traceable ring signature algorithm into a smart contract and deploy it on Ethereum. When the system detects information that disrupts normal traffic, the smart contract will trace the sender of the information and return it to TA.
5. We develop a mechanism to punish malicious users. For users who send messages that disrupt normal traffic, TA will determine punishment measures based on the number of times the user has done evil to ensure other vehicles' safety when sharing information.

This paper is organized as follows: In Section 2, we review some related works on data security and privacy protection for VANETs. In Section 3, we introduce the relevant technologies used in the scheme. In Section 4, we present the system overview. In Section 5, we describe the proposed security traffic data sharing scheme based on blockchain and traceable ring signature for VANETs and perform correctness. In Section 6, we perform a secure analysis of the scheme. In Section 7, we evaluate the performance of the scheme and present the simulation results. In Section 8, we draw conclusions of this article.

2 Related works

In recent years, more and more scholars have devoted themselves to the research of VANETs to address the security issues of data storage and sharing in VANETs as well as the privacy protection of drivers. To protect the sensitive information transmission and authentication in VANETs, suitable public key infrastructures (PKI) was embedded [20, 21] in real-time, and a certification authority (CA) issues anonymous certificates to hide the real identity of vehicles during the communication process. As PKI cannot provide location privacy and achieve an equitable distributed revocation mechanism, Wasef et al. [22] introduced random encryption periods to protect the location privacy of vehicles and proposed an efficient decentralized revocation protocol, which enables a group of neighboring vehicles to revoke malicious vehicles in their vicinity. Benarous et al. [23] pointed out the existing PKI infrastructure is centralized, then put forward a blockchain-based pseudonym management framework for VANETs. Those schemes prevented malicious vehicles from entering VANETs, minimized the cost of certificate and signature verification, and designed a tracking mechanism to revoke vehicles that behave continuously in VANETs. However, PKI-based authentication schemes all have similar shortcomings: 1) need a trusted CA to issue certificates; 2) high computational cost of certificate and signature verification; 3) certificate storage and key management are difficult.

Several researchers have proposed ID-based privacy preserving authentication schemes to solve the problems encountered in PKI-based schemes in VANET applications, these schemes reduce communication costs effectively. In 1984, Shamir [24] proposed an ID-based cryptosystem, where the user's public key exists by itself (e.g., the user's ID, email address, etc.) and the key generation center generates the corresponding private key based on the user's identity information and transmits it to the user. Deng et al. [25] pointed out current ID-based two-party authenticated key agreement schemes are not necessarily safe in real life. To accomplish the energy-efficient privacy of communicators and security of communication, Akram et al. [26] designed an identity-based authentication system for vehicular cloud computing and it also uses radio frequency identification. Since ID-based authentication schemes rely on CAs to generate private keys based on users' information, key management is considered the cornerstone of the security framework in VANETs and has become the focus of researchers' research. Lei et al. [27] and Ma et al. [28] presented a corresponding framework for secure key management, which combines the distributed idea of blockchain to design a key management scheme suitable for VANETs. Malhi et al. [29] proposed an efficient privacy-preserving scheme using an aggregated signature verification scheme. Rasheed et al. [30] put forward a group-based adaptive zero-knowledge proof authentication protocol that is lightweight and supports users for trade-off selection. Li et al. [31] and Coruh and Bayat [32] implemented an authentication scheme for VANETs with a revocation mechanism. Given the respective advantages of PKI-based and ID-based schemes, Wang et al. [33] combined the advantages of both schemes and proposed a hybrid conditional privacy-preserving authentication protocol for VANETs. The protocol based on the PKI certificate and identity-based signature to achieve the goal of user authentication.

Recently, the natural features of blockchain such as decentralization, immutability and traceability have attracted the interest of researchers, and many of them use blockchain to solve the problems of privacy protection, identity authentication and secure transmission in VANETs. Based on the current technical challenges of VANETs, Li et al. [34] designed a novel decentralized VANETs architecture using the natural advantages of blockchain, thereby avoiding centralization and entities' mutual distrust in VANETs. Addressing the security of transmissions in VANETs and the collection of private driver information, according to Shrestha et al. [35], a new blockchain can solve the problem of information security exchange by creating a local blockchain with national boundaries. Feng et al. [36] and Lin et al. [37] used blockchain technology for privacy-preserving authentication of VANETs to achieve conditional privacy protection by exploiting the traceable nature of blockchain. The corresponding revocation mechanism is designed in their proposed scheme, and the registration information can be revoked by the TA for misbehaving vehicles. Gong et al. [38] constructed a blockchain privacy protection scheme based on ring signature. Through this scheme, data authorized to be shared in the IoT is transmitted through the system in the form of ciphertext, and the identity information of the data sender is protected. However, no tracking mechanism was designed into the scheme. If vehicles in VANETs send false messages, the TA cannot identify the specific source through the signature on the message. Therefore, utilizing blockchain technology for traceability can achieve data security protection and trusted resource sharing.

3 Preliminaries

3.1 Bilinear pairing

We denote three (multiplicative) cyclic groups G_1, G_2 and G_T with the same order q generated by the generating element P. Let's denote g_1 be a generator of G_1, g_2 be a generator of G_2. The bilinear pairing $e : G_1 \times G_2 \rightarrow G_T$, need to satisfy the following three properties:

- **Bilinearity:** For all $a, b \in Z_q^*$ and $g_1 \in G_1, g_2 \in G_2$,there is $e(ag_1, bg_2) = e(g_1, g_2)^{ab}$.
- **Non-degeneracy:** $\exists g_1 \in G_1$, $\exists g_2 \in G_2$, there is $e(g_1, g_2) \neq 1_{G_T}$.
- **Computability:** For all $g_1 \in G_1, g_2 \in G_2$, that can easily compute the bilinear pairing $e(g_1, g_2)$.

3.2 Ring signature

In 2001, the concept of ring signature was first introduced by Rivest et al. [39]. A ring signature can protect the privacy of the signer by specifying a set of possible signers without revealing the identity of the actual signer. We refer to a set of possible signers as "ring", and the ring member that generates the actual signature as "signer". The signer signs the message with his own private key and the public keys of other ring members, and the other members of the ring are called "non-signer". We assume that there are n possible signers in a set, the ring signature scheme is defined as follows [40]:

1. ***Keygen:*** A probabilistic polynomial time algorithm, input safety parameter κ, system parameters $param = \{x_i, y_i\}$, where x_i,y_i are the private key and public key of ring member u_i. Different driver's public and private key may come from different PKI.
2. ***Ringsign:*** A probabilistic polynomial time algorithm, user u_i inputs the message m and the public keys $L = y_1, y_2, \cdots, y_n$ of the other ring members and his private key x_i, output a signature R to the message m, certain parameters in the signature form a ring according to certain rules.
3. ***Ringverify:*** A deterministic algorithm, input (m, R), if R is the ring signature of message m, output "*True*", otherwise, output "*False*".

Ring signatures, when first proposed by Rivest et al, got their name from the fact that a certain hidden parameter in the signature forms a ring according to certain rules. Subsequently, many schemes proposed by researchers do not require that certain parameters in the signature need to form a ring, as long as the formation of a signature satisfies the spontaneity, anonymity and group properties, it is also called a ring signature.

3.3 Blockchain and smart contract

Blockchain is the core data storage structure of our proposed scheme. It is essentially a distributed ledger in which all transactions are not easily tampered with, and has the characteristics of decentralization, immutability and traceability. The data in VANETs are recorded in blocks in chronological order, and then the newly generated blocks are appended to the blockchain by a consensus algorithm. Depending on the permission settings, blockchains can be divided into public blockchain, consortium blockchain and private blockchain. In our proposed scheme, we choose Ethereum as the underlying network architecture, and any vehicle registered through TA can access the VANETs data stored in Ethereum. In addition, Ethereum can efficiently process transactions and support a Turing-complete smart contract.

The smart contract is a computerized transactional protocol that enforces the terms of a contract. As embedded in the blockchain, it enables the agreement to be executed automatically without the intervention of a trusted third party. In general, it offers some appealing features when combined with blockchain, such as automatic execution, immutability, and decentralization. In our scheme, we use smart contract to track illegal vehicles in VANETs, providing the corresponding application binary interface (ABI) to access the TA's secure database, and the ABI supports querying the user's tracking private key. When a traffic-disrupting message is found in VANETs, the smart contract is automatically executed to track down the real signer and then submit the relevant information to the TA. Moreover, we require that only the smart contract can determine the real signer. The TA decides how to dispose of these illegal vehicles, and then records the result in the blockchain.

3.4 Edge computing

Edge computing [41] is a novel computing paradigm for performing computational tasks at the edge of the network that emphasizes being closer to the user and closer to the data source. Edge computing can provide low-latency computing, high-speed caching and location-aware services for vehicles, and can enable real-time information interaction for VANETs. Nowadays, the main schemes to implement vehicle edge computing (VEC) include mobile edge computing and fog computing in VANETs. Mobile edge computing is a promising approach to deploy computationally intensive and time-sensitive tasks in VANETs where computational resources are provided to vehicles by an enhanced network infrastructure and network routers are upgraded by deploying hardware devices with more computational power. Fog computing in VANETs, on the other hand, treats vehicles as infrastructure that performs a large number of computing tasks close to the drivers. It makes full use of the idle communication and computing power of the vehicles in the network to maximize the resources of these vehicles.

In order to facilitate the network functions of VANETs and improve traffic efficiency, the VEC technology is introduced in our proposed scheme. The addition of VEC can offload tasks such as message authentication, identity tracking and user revocation from systems to edge computing through enhanced RSUs to reduce latency in VANETs and enable real-time interaction of information. Only operations related to driver privacy need to be performed in OBU, e.g., key generation, message signing. When a vehicle needs to synchronize data, it can access the blockchain to obtain the information.

4 System overview

In this section, we introduce the system model, security model and scheme framework used in our proposed scheme.

4.1 System model

The system model of the proposed scheme mainly consists of seven entities: trusted authority (TA), vehicles, RSUs, traffic management department (TMD), edge computing, blockchain, and smart contract. A complete model of our system is shown in Fig. 1.

1. ***Trusted authority (TA):*** TA has powerful computing and storage capabilities and is responsible for the daily maintenance of VANETs. For other entities in VANETs, TA is considered fully credible and uncompromising to any adversary. TA is also responsible for the initialization of VANETs, generating system parameters and providing registration to vehicles. When vehicles want to join VANETs, they need to submit registration applications to TA first, and only vehicles with approved applications can join VANETs. When a vehicle sends a false message to disturb the normal traffic order, TA will track the signed vehicle based on the digital signature in the message and hand over the vehicle information to the TMD for penalty. TA will revoke the registration information of the vehicle when the vehicle is repeatedly misbehaving.
2. ***Vehicle:*** Vehicles in VANETs are equipped with OBU and TPD. Vehicles are considered to be data collection devices in VANETs, where vehicles can communicate with other vehicles and RSUs to share road messages through OBU. When a vehicle applies for registration with TA, TA will preload the initialization parameters in OBU. The TPD in OBU will ensure that these parameters are secure and will not be tampered with by attackers at will.
3. ***RSU:*** RSUs are installed on both sides of the road and support the DSRC protocol, allowing communication with vehicles within a specific range. Specifically, RSUs can receive messages from vehicles, validate them, forward the validated messages to other vehicles and store them in the blockchain. The RSUs are interconnected and they exchange messages over a secure wired network.
4. ***Traffic management department (TMD):*** The traffic management department manages the vehicles in VANETs. It receives drivers' information from TA, penalizes the misbehaving vehicles accordingly, and records the results of the penalties, through node consensus, in the blockchain.
5. ***Edge computing:*** With powerful computing and storage capabilities, edge computing processes data at the edge of the network, reducing network latency while improving data security and privacy. Edge computing enhances the efficiency of VANETs by accessing RSUs, TAs and other entities in VANETs through edge nodes and offloading most of the data computation to cloud servers through edge nodes.
6. ***Blockchain:*** We use the public blockchain as the decentralized underlying architecture for our scheme to instantiate VANETs. It is verified by RSUs based on consensus algorithms for messages, and registered vehicles can

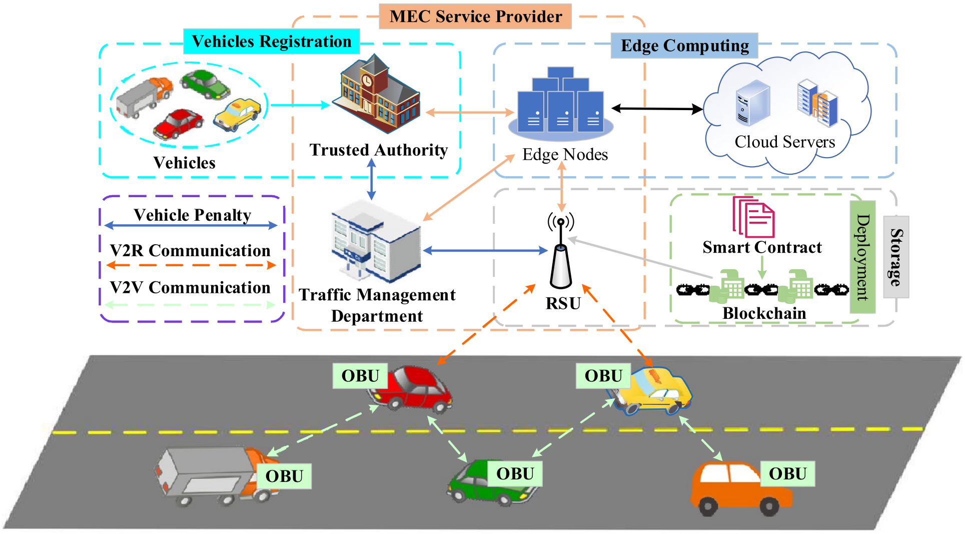


Fig. 1 System model of our scheme

access the blockchain through RSUs to get the required traffic data.

7. ***Smart contract:*** Smart contract is a computer protocol that is developed by TA and deployed on a blockchain. It accesses the TA's security database, obtains the drivers' tracking private key, calculates T_i, and then determines the real signer by bilinear pairing operations. When an event triggers a clause in the contract, the contract is automatically executed to track down the real signer and return the result to the TA.

4.2 Security model

A secure and efficient ring signature privacy protection authentication scheme needs to satisfy both anonymity and unforgeability. According to Bender et al.'s security definition of anonymity and unforgeability of different strength ring signature [42], we study the definition of security models for recent privacy-preserving authentication schemes for VANETs. Combined with more secure traceability, we propose the traffic data security sharing scheme. The scheme's security model should meet four security requirements: unforgeability, anonymity, traceability, and resistance to cyberattacks:

- **Unforgeability:** Unforgeability means that no one other than a ring member can generate a legitimate ring signature for the identity set $\mathcal{L}$ unless it has access to the corresponding private key of the ring member. Even if an attacker can obtain the signature of a message from a random oracle that generates a ring signature, its probability of successfully forging a legitimate ring signature is negligible.
- **Anonymity:** After a vehicle is registered with TA as a legal user, the true identity of each vehicle is hidden from other entities in VANETs. Since the anonymity of ring signature is unconditionally anonymous, the attacker determines the real signer is negligible even if he obtains the private keys of all ring members.
- **Traceability:** In contrast to other ring signature schemes, this paper proposes a traceable ring signature scheme where the anonymity of the driver is conditionally anonymous. TA can obtain information about vehicles that misbehave (such as driving illegally, sending false information, etc.) and penalize them accordingly. TA will revoke the registration information of vehicles that have repeatedly done evil.
- **Resistance to cyberattacks:** Our proposed scheme can resist common network attacks such as distributed denial of service (DDoS) attack, replay attack, false message attack, and vehicle impersonation attack. In existing VANETs, our proposed scheme works well.

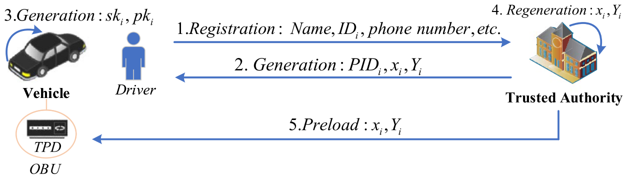

3.Generation : sk_i, pk_i
1.Registration : Name, ID_i, phone number, etc.
4. Regeneration : x_i, Y_i
Vehicle
Driver
2. Generation : PID_i, x_i, Y_i
Trusted Authority
5.Preload : x_i, Y_i
TPD
OBU

Fig. 2 Vehicle registration and key generation

4.3 Scheme framework

In this section, we focus on how our proposed scheme can protect vehicle privacy in VANETs. In our VANETs model, the TA is mainly responsible for the initialization of the system and the generation of vehicle tracking key, the vehicle registration and key generation of which is shown in Fig. 2. Any vehicle that wishes to join VANETs needs to submit a registration application to TA, and TA will generate a pseudonym for the vehicle, which is used for vehicle communication in VANETs.

When a vehicle enters the RSU area, the vehicle sends a ring request to the RSU, and the RSU stores its public key pk_i in the ring set R and its tracking public key in the set Y. Once the vehicle needs to send a message, it randomly selects a subset R_1 of n valid public keys and the corresponding tracking public key Y' to sign the message. Fig. 3 displays the process of generating and verifying signature.

The system detects a disturbance of normal traffic order or a false message, smart contract traces the signer through the signature in the message, and then the TA executes the corresponding penalty. When the number of vehicle misbehaviors is less than the system setting, TA will submit the vehicle information to the TMD, which will penalize the vehicle and record it on the blockchain. TA will revoke the vehicle registration once the number of misbehaviors reaches a certain level. The tracking and penalty for the vehicle is shown in Fig. 4.

The details of the system framework are as follows:

1. TA executes ***System setup*** to generate system parameters, conducts blockchain deployments, and deploys

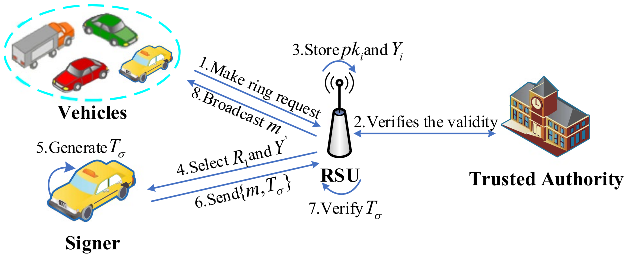
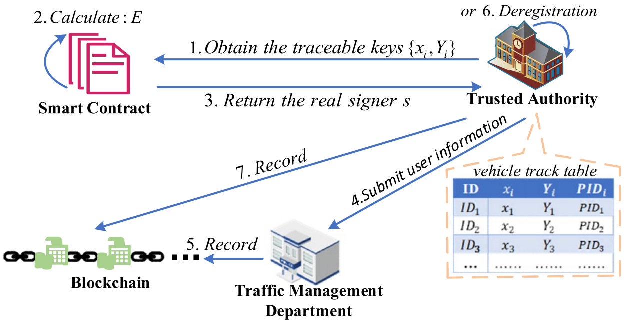


Fig. 3 Signature generation and verification


2. Calculate : E
or 6. Deregistration
1. Obtain the traceable keys {x_i, Y_i}
Smart Contract
3. Return the real signer s
Trusted Authority
7. Record
4.Submit user information
vehicle track table
5. Record
Blockchain
Traffic Management Department

ID	x_i	Y_i	PID_i
ID_1	x_1	Y_1	PID_1
ID_2	x_2	Y_2	PID_2
ID_3	x_3	Y_3	PID_3
...	......	......	......

Fig. 4 Vehicle tracking and penalty

Ethereum in VANETs. Then, TA performs ***Smart contract deployment*** to deploy the smart contract. Smart contract deployed in the blockchain are given a unique address, and are automatically executed when reach the set conditions.

2. Vehicles that want to be added to VANETs need to submit a registration application to TA using their real identity. TA executes ***Vehicle registration*** to generate a pseudonym PID_i for each vehicle.
3. After successful registration, TA via ***Traceable key generation*** to produce tracking key $\{x_i, Y_i\}$, which are used for tracing of real identity. The user performs ***Vehicle key generation*** through the OBU to generate private-public key $\{sk_i, pk_i\}$ for vehicles.

Table 1 Key symbol and definition

Symbol	Definition
TA	Trust Authority
OBU	On Board Unit
RSU	Roadside Unit
λ	System's security parameter
q	Large prime
P	Generator of G_1
G_1, G_T	Multiplicative cyclic group
V_i	Vehicle
MSK	TA's master secret key
MPK	TA's master public key
e	Bilinear pairing operation
$H_0 \cdots H_5$	Hash functions
ID_i	Vehicle ID
PID_i	V_i's pseudonym
x_i	V_i's tracking private key
Y_i	V_i's tracking public key
sk_i	V_i's private key
pk_i	V_i's public key
R_1	Public key set
Y'	Tracking public key set
M	Pseudonym set

4. When entering the coverage area of the RSU, the vehicle makes a ring request to the RSU. After receiving the request information, RSU verifies the validity and timeliness to the TA, and then stores the user public key in the ring set $R = \{pk_1, pk_2, \cdots, pk_{\max}\}$, corresponding to the tracking key set $Y = \{Y_1, Y_2, \cdots, Y_{max}\}$, the ring set and the tracking key set are dynamically updated as vehicles enter and leave the coverage area of the RSU. When the vehicle needs to sign a message $m \in \{0,1\}^*$, the vehicle randomly selects a subset $R_1 = \{pk_1, pk_2, \cdots, pk_n\} \in R$ and its corresponding subset of tracking key $Y' = \{Y_1, Y_2, \cdots, Y_n\} \in Y$ from the RSU. The vehicle performs ***Signature generation*** to output T_σ. When RSU receives the message m, they execute ***Signature verification***. Receive the message if it is valid, otherwise, reject it.
5. When the message is invalid, the smart contract performs ***Vehicle tracking*** to track specific vehicles and report the results to TA over a secure channel. TA executes ***Vehicle discipline and revocation*** to ensure traffic safety.

5 Proposed traffic data sharing scheme

In this section, we design a security traffic data sharing scheme based on blockchain and traceable ring signature for VANETs and verify its correctness. Table 1 is the symbol description involved in our scheme.

5.1 Design of our scheme

This scheme mainly comprises of the following night parts: system setup, smart contract deployment, vehicle registration, traceable key generation, vehicle key generation signature generation, signature verification, signer tracking, and signer discipline and revocation. Fig. 5 is the flowchart of our proposed scheme.

1. ***System setup:*** Input security parameter λ, TA initializes the system as follows:
    1. TA chooses two multiplicative cyclic groups G_1, G_T with same order q, where q is a large prime. Let's consider P be a generator of G_1, and the bilinear pairing $e : G_1 \times G_1 \rightarrow G_T$.
    2. TA select random number $MSK = a \in Z_q^*$ as its master secret key and compute its master public key $MPK = aP$.
    3. TA pick six general one-way hash functions $H_0 : \{0,1\}^* \rightarrow Z_q^*$, $H_1 : \{0,1\}^* \times Z_q^* \rightarrow G_1$, $H_2 :$ $H_2 : \{0,1\}^* \times \{0,1\}^* \rightarrow Z_q^*$, $H_3 : Z_q^* \times G_1 \rightarrow Z_q^*$, $H_4 : \{0,1\}^* \times Z_q^* \rightarrow Z_q^*$, $H_5 : \{0,1\}^* \times Z_q^* \times \cdots \times Z_q^* \rightarrow Z_q^*$.

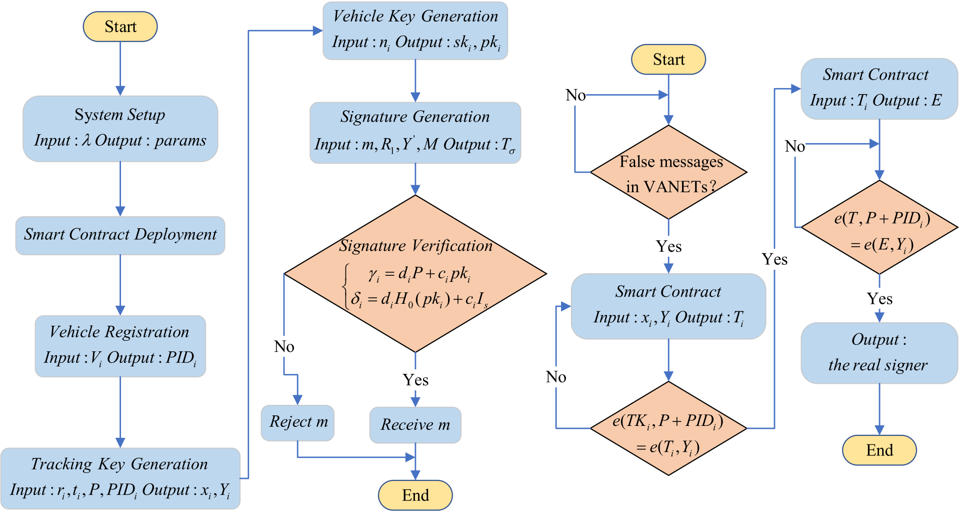

Start
System Setup
$Input : \lambda\ Output : params$
Smart Contract Deployment
Vehicle Registration
$Input : V_i\ Output : PID_i$
Tracking Key Generation
$Input : r_i, t_i, P, PID_i\ Output : x_i, Y_i$
Vehicle Key Generation
$Input : n_i\ Output : sk_i, pk_i$
Signature Generation
$Input : m, R_1, Y', M\ Output : T_\sigma$
Signature Verification
$\begin{cases} \gamma_i = d_iP + c_ipk_i \\ \delta_i = d_iH_0(pk_i) + c_iI_s \end{cases}$
No
Yes
Reject m
Receive m
End
Start
No
False messages in VANETs?
Yes
Smart Contract
$Input : x_i, Y_i\ Output : T_i$
No
$e(TK_i, P + PID_i) = e(T_i, Y_i)$
Yes
Smart Contract
$Input : T_i\ Output : E$
No
$e(T, P + PID_i) = e(E, Y_i)$
Yes
Output : the real signer
End

Fig. 5 Flowchart of our scheme

4. Finally, TA keeps MSK, as well as public the system parameters $params = \{q, P, e, G_1, G_T, MPK, H_0, H_1, H_2, H_3, H_4, H_5\}$.

2. ***Smart contract deployment:*** TA takes inputs of track smart contract drafts, compiles, and deploys them into the blockchain. After being verified by blockchain, track smart contract get their unique addresses and when a vehicle in VANETs sends a false message, the smart contract will automatically track the sender of the message and submit its real identity to the TA.
3. ***Vehicle registration:*** Every vehicle V_i that wants to join VANETs submits its real identity to the TA for registration, and TA generates the pseudonym PID_i for V_i as following steps:

    1. Vehicle V_i randomly choose a constant $l_i \in Z_q^*$, and computes $M_i = l_iP, N_i = l_iMPK, OID_i = ID_i \oplus N_i$. Then, V_i sends $\{M_i, OID_i\}$ to TA.
    2. After receiving $\{M_i, OID_i\}$, TA computes $ID_i = OID_i \oplus N_i = OID_i \oplus (l_iaP) = OID_i \oplus aM_i$. If vehicle's ID_i is legal, TA will generate the pseudonym for vehicle, it computes $PID_i = H_1(a \| OID_i) \oplus ID_i$ and submit PID_i to vehicle.
    3. V_i preloads PID_i into OBU.

4. ***Tracking key generation:*** In order to track vehicle's real identity in case of a dispute, TA generates the tracking key Y_i for each successfully registered vehicle V_i as following algorithm:

    1. TA randomly picks a constant $r_i \in Z_q^*$ and computes $x_i = H_2(r_i \| t_i)$, where t_i is the validity period of constant x_i.
    2. Then TA computes $Y_i = x_i(P + PID_i)$, and it secretly sends $\{x_i, Y_i\}$ to vehicle V_i through a secure channel.
    3. V_i preloads $\{x_i, Y_i\}$ into OBU, and Y_i as its tracking public key.
    4. TA stores $\{x_i, Y_i, PID_i\}$ in its security database.

5. ***Vehicle key generation:*** Vehicle V_i randomly selects a number $n_i \in Z_P^*$ and computes its private key $sk_i = H_3(n_i \| PID_i)$, $pk_i = sk_iP$.
6. ***Signature generation:*** Assuming that the signer in the system is s, the message to be signed is $m \in \{0, 1\}^*$, s selects a public key set $R_1 = \{pk_1, pk_2, \cdots, pk_n\}$ of n drivers in the system, and the corresponding tracking public key set $Y' = \{Y_1, Y_2, \cdots, Y_n\}$ and the corresponding pseudonym set $M = \{PID_1, PID_2, \cdots, PID_n\}$, then generate a signature on message m by following steps.

    1. For each public key pk_i generate correspond attribute values L_i, K_i, s randomly picks different number $u_i, v_i \in Z_q^*$, and computes the following:

    $$L_i = \begin{cases} u_sP & i = s \\ u_iP + v_ipk_i & i \neq s \end{cases} \tag{1}$$

    $$K_i = \begin{cases} u_sH_0(pk_s) & i = s \\ u_iH_0(pk_i) + v_iI_s & i \neq s \end{cases} \tag{2}$$

    Among them: $I_s = sk_sH_o(pk_S)$, which is a signature image of the message m, is used to prevent double spending attacks in the system.
    2. Signer s calculate $h = H_4(m \| R_1)$, and generates partial signatures c_i, d_i of message m by following algorithm:

$$c_i = \begin{cases} H_5(h, L_1, \cdots, L_n, K_1, \cdots K_n) - \sum_{i=1, i \neq s}^{n} c_i & i = s \\ v_i & i \neq s \end{cases} \tag{3}$$

$$d_i = \begin{cases} u_s - c_s sk_s & i = s \\ u_i & i \neq s \end{cases} \tag{4}$$

3. Signer s selects a random number $w_i \in Z_q^*$ for generating partial signatures that tracing signer's real identity. The detailed steps are described as follows:

$$T_i = w_i(P + PID_i)(i = 1, 2, \cdots, n) \tag{5}$$

$$TK_i = w_i Y_i(i = 1, 2, \cdots, n) \tag{6}$$

$$T = x_s \sum_{i=1}^{n} T_i \tag{7}$$

4. The traceable signature of message m signed by signer s output as:

$$T_\sigma = (I_s, c_1, c_2, \cdots, c_n, d_1, d_2, \cdots, d_n, TK_1, TK_2, \cdots, TK_n, T) \tag{8}$$

5. Finally, s sends the message $\{m, T_\sigma\}$ to the RSU.

7. ***Signature verification:*** The signature verification of message m is done by RSUs, they can obtain the public keys of all members of the ring signature, and verify the signature T_σ output by signer s by the following:

$$\begin{cases} \gamma_i = d_i P + c_i pk_i \\ \delta_i = d_i H_0(pk_i) + c_i I_s \end{cases} \tag{9}$$

Determine whether the above formula is valid, if the formula validation passes, receive the message m and record it in the newly generated block, otherwise, it refuses to receive the message m.

8. ***Signer tracking:*** When a false message sent by a malicious vehicle is detected, the smart contract deployed in the blockchain identifies the signer s of the faked message through the traceable ring signature T_σ in the message.

   1. The smart contract accesses the TA's security database through an interface to obtain the traceable keys $\{x_i, Y_i\}$ of all ring members, and $T_i = TK_i \cdot x^{-1}$ is computed.
   2. When the T_i value is obtained, verify the validity of T_i by bilinear mapping $e(TK_i, P + PID_i) = e(T_i, Y_i)$. If all T_i are valid, then compute $E = \sum_{i=1}^{n} T_i$.
   3. After the above steps, the real signer can be determined $e(T, P + PID_i) = e(E, Y_i)$.

9. ***Signer discipline and revocation:*** After determining the true signer s of the invalid message, the smart contract submits the vehicle ID, signer and other information to the TA via a secure channel, and the TA will decide the penalties for vehicle who sends invalid message. If the vehicle has only just started sending false message, TA will submit its relevant information to the TMD, it agency to impose penalties on vehicle, penalty results are recorded in the newly generated block, it will be added to the blockchain after passing the consensus algorithm verification. If the vehicle repeatedly sends inaccurate messages into VANETs, the TA revokes the vehicle's registration information, and they cannot access any data in VANETs. To ensure the security of VANETs, vehicles deregistered by TA for malicious acts cannot be re-registered for a certain period of time. During the re-registration process, TA will conduct a strict review of the registration application submitted by the vehicle, and only those who pass the review will be re-registered into VANETs.

Algorithm 1 Vehicle Tracking

```
Input: Signature $T_\sigma$
Output: The real signer $s$
for smart contract do
    Get $x_i$ from TA's security database;
    Calculate $T_i \leftarrow TK_i \cdot x^{-1}$;
    if $e(TK_i, P + PID_i) = e(T_i, Y_i)$ then
        Calculate $E \leftarrow \sum_{i=1}^{n} T_i$;
        if $e(T, P + PID_i) = e(E, Y_i)$ then
            Return real signer $s$;
        else
            Return failed to track the signer;
        end if
    else
        Return Calculate $T_i$;
    end if
end for
```

5.2 Proof of correctness

The correctness of signature T_σ in message m is proved as follows:

$$\sum_{i=1}^{n} c_i = H_5(h, \gamma_1, \gamma_2, \cdots \gamma_n, \delta_1, \delta_2, \cdots, \delta_n) \tag{10}$$

when $i \neq s$, the conversion of γ_i, δ_i is as follow:

$$\begin{aligned} \gamma_i &= d_i P + c_i pk_i \\ &= u_i P + v_i pk_i \\ &= L_i \end{aligned} \tag{11}$$

$$\begin{aligned} \delta_i &= d_i H_0(pk_i) + c_i I_s \\ &= u_i H_0(pk_i) + v_i I_s \\ &= K_i \end{aligned} \tag{12}$$

when $i = s$, the conversion of γ_i, δ_i is as follow:

$$\begin{aligned}\gamma_s &= d_sP + c_spk_s \\ &= (u_s - c_ssk_s)P + c_spk_s \\ &= u_sP \\ &= L_s\end{aligned} \tag{13}$$

$$\begin{aligned}\delta_s &= d_sH_0(pk_s) + c_sI_s \\ &= (u_s - c_ssk_s)H_0(pk_s) + c_ssk_sH_0(pk_s) \\ &= u_sH_0(pk_s) \\ &= K_s\end{aligned} \tag{14}$$

Thus, the above relationships can be used to verify the correctness of the traceable ring signature scheme proposed in this scheme:

$$\begin{aligned}&H_5(h, \gamma_1, \gamma_2, \cdots, \gamma_n, \delta_1, \delta_2, \cdots, \delta_n) \\ &= H_5(h, L_1, L_2, \cdots, L_3, K_1, K_2, \cdots, K_3) \\ &= c_s + \sum_{i=1, i\neq s}^{n} c_i \\ &= \sum_{i=1}^{n} c_i\end{aligned} \tag{15}$$

In the trace signer stage, the smart contract will access the TA's security database and calculate T_i, which must be verified by the formula $e\left(TK_i, P + PID_i\right) = e(T_i, Y_i)$, the proof of correctness is as below:

$$\begin{aligned}e(TK_i, P + PID_i) &= e(w_iY_i, P + PID_i) \\ &= e(w_ix_i(P + PID_i), P + PID_i) \\ &= e(w_i(P + PID_i), x_i(P + PID_i)) \\ &= e(T_i, Y_i)\end{aligned} \tag{16}$$

After T_i has been verified, the signer is be found by formula $e\left(T, P + PID_i\right) = e(E, Y_i)$, the proof of correctness is as below:

$$\begin{aligned}e(T, P + PID_i) &= e(x_i\sum_{i=1}^{n} T_i, P + PID_i) \\ &= e(\sum_{i=1}^{n} T_i, x_i(P + PID_i) \\ &= e(E, Y_i)\end{aligned} \tag{17}$$

6 Security analysis

According to the security model given in Section 4 of this paper, we present a provable security analysis of our proposed scheme.

6.1 Unforgeability

Theorem 1: The proposed scheme is unforgeability.

Proof: In the random oracle model, the attacker $\mathcal{A}$ can adaptively choose a message to attack. $\mathcal{S}$ is a challenger who can use $\mathcal{A}$'s ability to solve Elliptic Curve Discrete Logarithm Problem (ECDLP). Assume that attacker $\mathcal{A}$ attacks the scheme with a non-negligible probability, and asks a series of queries to challenger $\mathcal{S}$. Given $P, Q = aP$, $\mathcal{S}$'s goal is to output a scheme s of ECDLP by interacting with $\mathcal{A}$. For this, $\mathcal{S}$ chooses PID_i^* as the challenge anonymous-identity, the interaction between challenger $\mathcal{S}$ and attacker $\mathcal{A}$ is shown below:

1. ***System setup:*** Given a security parameter λ, $\mathcal{S}$ initializes system parameters $params = \left\{q, P, e, G_1, G_T, MPK, H_0, H_1, H_2, H_3, H_4, H_5\right\}$ by running system setup algorithm and sets $MPK = Q = aP$. Then, keep a as its private key and sends $params$ to $\mathcal{A}$.
2. H_0 ***queries:*** $\mathcal{S}$ maintains a list $L_{H_0} : (pk_i, h_{i0})$, which is set empty in the initial stage. After receiving the query on $H_0(pk_i)$, $\mathcal{S}$ look up it in the list L_{H_0}. If it exits, $\mathcal{S}$ returns $h_{i0} \in Z_q^*$ to $\mathcal{A}$. Otherwise, $\mathcal{S}$ randomly selects a number $h_{i0} \in Z_q^*$ as well as sets $h_{i0} = H_0(pk_i)$ and returns h_{i0} to $\mathcal{A}$. Finally, $\mathcal{S}$ adds the tuple (pk_i, h_{i0}) in the list L_{H_0}.
3. H_1 ***queries:*** $\mathcal{S}$ maintains a list $L_{H_1} : (a, OID_i, h_{i1})$, which is set empty in the initial stage. After receiving the query on (a, OID_i), $\mathcal{S}$ look up it in the list L_{H_1}. If it exits, $\mathcal{S}$ returns $h_{i1} \in G_1$ to $\mathcal{A}$. Otherwise, $\mathcal{S}$ randomly selects a number $h_{i1} \in G_1$ as well as sets $h_{i1} = H_1(a\|OID_i)$ and returns h_{i1} to $\mathcal{A}$. Finally, $\mathcal{S}$ adds the tuple (a, OID_i, h_{i1}) in the list L_{H_1}.
4. H_2 ***queries:*** $\mathcal{S}$ maintains a list $L_{H_2} : (r_i, t_i, h_{i2})$, which is set empty in the initial stage. After receiving the query on (r_i, t_i), $\mathcal{S}$ look up it in the list L_{H_2}. If it exits, $\mathcal{S}$ returns $h_{i2} \in Z_q^*$ to $\mathcal{A}$. Otherwise, $\mathcal{S}$ randomly selects a number $h_{i2} \in Z_q^*$ as well as sets $h_{i2} = H_2(r_i\|t_i)$ and returns h_{i2} to $\mathcal{A}$. Finally, $\mathcal{S}$ adds the tuple (r_i, t_i, h_{i2}) in the list L_{H_2}.
5. H_3 ***queries:*** $\mathcal{S}$ maintains a list $L_{H_3} : (n_i, PID_i, h_{i3})$, which is set empty in the initial stage. After receiving the query on (n_i, PID_i), $\mathcal{S}$ look up it in the list L_{H_3}. If it exits, $\mathcal{S}$ returns $h_{i3} \in Z_q^*$ to $\mathcal{A}$. Otherwise, $\mathcal{S}$ randomly selects a number $h_{i3} \in Z_q^*$ as well as sets $h_3 = H_3(n_i\|PID_i)$ and returns h_{i3} to $\mathcal{A}$. Finally, $\mathcal{S}$ adds the tuple (n_i, PID_i, h_{i3}) in the list L_{H_3}.
6. H_4 ***queries:*** $\mathcal{S}$ maintains a list $L_{H_4} : (m, R_1, h_{i4})$, which is set empty in the initial stage. After receiving the query on (m, R_1), $\mathcal{S}$ look up it in the list L_{H_4}. If it exits, $\mathcal{S}$ returns $h_{i4} \in Z_q^*$ to $\mathcal{A}$. Otherwise, $\mathcal{S}$ randomly selects a number $h_{i4} \in Z_q^*$ as well as sets $h_{i4} = H_4(m\|R_1)$ and returns h_{i4} to $\mathcal{A}$. Finally, $\mathcal{S}$ adds the tuple (m, R_1, h_{i4}) in the list L_{H_4}.
7. H_5 ***queries:*** $\mathcal{S}$ maintains a list $L_{H_5} : (h, L_1, \cdots, L_n, K_1, \cdots, K_n, h_{i5})$, which is set empty in the initial stage.

After receiving the query on $(h, L_1, \cdots, L_n, K_1, \cdots, K_n)$, $\mathcal{S}$ look up it in the list L_{H_5}. If it exits, $\mathcal{S}$ returns $h_{i5} \in Z_q^*$ to $\mathcal{A}$. Otherwise, $\mathcal{S}$ randomly selects a number $h_{i5} \in Z_q^*$ as well as sets $h_{i5} = H_5(h, L_1, \cdots, L_n, K_1, \cdots, K_n)$ and returns h_{i5} to $\mathcal{A}$. Finally, $\mathcal{S}$ adds the tuple $(h, L_1, \cdots, L_n, K_1, \cdots, K_n, h_{i5})$ in the list L_{H_5}.

8. ***Vehicle public key queries:*** $\mathcal{S}$ maintains a list $L_{pk} : (PID_i, r_i, Y_i, n_i, pk_i)$. This list is initially empty. When the attacker $\mathcal{A}$ make a public key query to the challenger $\mathcal{S}$, $\mathcal{S}$ looks up it in the list L_{pk}. If it exits, $\mathcal{S}$ return (Y_i, pk_i) to $\mathcal{A}$. Otherwise, $\mathcal{S}$ recovers the tuple (r_i, t_i, h_{i2}) from L_{H_2}, and also recovers the tuple (n_i, PID_i, h_{i3}) from L_{H_3}, respectively. Then, it randomly selects two constant $r_i, n_i \in Z_p^*$ and calculates $x_i = H_2\left(r_i \| t_i\right)$, $Y_i = x_i\left(P + PID_i\right)$, $sk_i = H_1\left(n_i \| PID_i\right)$, $pk_i = sk_i P$. Finally, $\mathcal{S}$ set tuple (Y_i, pk_i) as its public key and sends it to $\mathcal{A}$, and also inserts the tuple $(PID_i, r_i, Y_i, n_i, pk_i)$ to L_{pk}.
9. ***Vehicle private key queries:*** When attacker $\mathcal{A}$ makes a public key query to $\mathcal{S}$, if $PID_i = PID_i^*$, $\mathcal{S}$ aborts this operation, otherwise $\mathcal{S}$ returns the corresponding private key (x_i, sk_i) to $\mathcal{A}$.
10. ***Signature queries:*** When $\mathcal{A}$ wants to query the signature on (PID_i, m), $\mathcal{S}$ check whether $PID_i = PID_i^*$. If $PID_i \neq PID_i^*$, $\mathcal{S}$ runs the signature generation algorithm to produce the corresponding signature T_σ and return it to $\mathcal{A}$. If $PID_i = PID_i^*$, $\mathcal{S}$ recovers tuple $(PID_i^*, r_i, Y_i^*, n_i, pk_i^*)$ from the list L_{pk}.Then, the attacker $\mathcal{A}$ uses the public key set R_1, the tracking public key set Y', and the pseudonym set M, randomly selects $u_i, v_i \in Z_q^*$, and then performs the following steps:

$$L_i = \begin{cases} u_s P & i = s \\ u_i P + v_i pk_i{}^* & i \neq s \end{cases} \tag{18}$$

$$K_i = \begin{cases} u_s H_0(pk_s{}^*) & i = s \\ u_i H_0(pk_i{}^*) + v_i I_s{}^* & i \neq s \end{cases} \tag{19}$$

$$h = (m \| R_1) \tag{20}$$

$$c_i = \begin{cases} H_5(h, L_1, \cdots, L_n, K_1, \cdots K_n) - \sum_{i=1, i \neq s}^{n} c_i & i = s \\ v_i & i \neq s \end{cases} \tag{21}$$

$$d_i = \begin{cases} u_s - c_s sk_s{}^* & i = s \\ u_i & i \neq s \end{cases} \tag{22}$$

$$T_i = w_i(P + PID_i{}^*)(i = 1, 2, \cdots, n) \tag{23}$$

$$TK_i = w_i Y_i{}^*(i = 1, 2, \cdots, n) \tag{24}$$

$$T = x_s \sum_{i=1}^{n} T_i^* \tag{25}$$

Finally, the traceable ring signature for message m is output as $T_{\sigma^*} = (I_s^*, c_1, c_2, \cdots, c_s^*, \cdots, c_n, d_1, d_2, \cdots, d_s^*, \cdots, d_n, TK_1, TK_2, \cdots, TK_s^*, \cdots, TK_n, T)$.

Forgery: The attacker $\mathcal{A}$ output the signature of another message m^* for the signer PID_i^*, with the similar construction, the $\mathcal{S}$ can get the same result, two valid ring signatures are output as T_σ and T_{σ^*}. The $\mathcal{S}$ output the result of $a = MSK$ as a scheme of the ECDLP. However, ECDLP is intractable, so both are contradictory. That is the scheme in this paper satisfies unforgeability.

6.2 Anonymity

Theorem 2: The signature is Anonymity of the signer in our proposed scheme.

Proof: The ring signature in this paper has the anonymity of the signer for any sets $R_1 = \left\{pk_1, pk_2, \cdots, pk_n\right\}$ and a random $pk_s \in R_1$, the probability Pr $\left[pk_i = pk_i^*\right] = 1/2$, where $T_\sigma = (I_s, c_1, c_2, \cdots, c_n, d_1, d_2, \cdots, d_n, TK_1, TK_2, \cdots, TK_n, T)$ is a traceable ring signature generate produced by pk_s.

During the communication process, vehicles use pseudonyms to hide their true identity as $PID_i = H_1(s \| OID_i) \oplus ID_i$, $OID_i = ID_i \oplus N_i$, and s is TA's master private key, which is stored in TA's secure database. The tracking key generated by TA is stored in the secure database, as $Y_i = x_i(P + PID_i)$, $x_i = H_1(r_i \| t_i)$, and t_i is the validity period of tracking key. When the tracking key expires, the TA regenerates the tracking key for the vehicle and sends it to the vehicle over a secure channel. The trace key stored in the OBU is overwritten by the newly generated. Except for TA, no adversary can obtain the true identity of the vehicle from the (PID_i, Y_i, pk_i). In the signature generate, the values of u_i, v_i are chosen at random from Z_q^*, and the number needed to generate the signer's private key is chosen randomly from Z_q^*. Therefore, the signature T_σ will not expose the information of the signer.

6.3 Traceability

Theorem 3: The proposed scheme achieves conditional traceability.

Proof: When false message m is found, smart contract can track the real identity of signer. Smart contract accesses the TA's security database to get the T_i value of the corresponding ring members, after getting all T_i values, the smart contract verifies their validity. Then traces the real signer by the tracking public key. Finally, it sends the signer's ID to the TA through secure channel. Only smart contract can access TA's secure database through the interface to get the corresponding parameters of the traceable ring signature members, the who satisfies the equation $\left(T, P + PID_i\right) = e\left(E, Y_i\right)$ is the real signer and it will send the result to the TA. Therefore, only TA can know the true identity of the signer, and no one else can trace the true identity of the signer.

6.4 Resistance to cyberattacks

Theorem 4: The proposed scheme can resist cyberattacks.

Proof: We designed scheme to inherit the resistance of blockchain technology to cyberattacks, which means that any modification to the smart contract can be prevented. Blockchain is distributed, and the system we designed uses blockchain as the underlying architecture, with the complete data required for the system stored in multiple nodes, system can function even if some points are down. Applying blockchain to system can eliminate single point of failure, effectively resist DDoS attacks and guarantee the overall security of the system.

Furthermore, tracking of the real signer is performed by a smart contract deployed in the blockchain, which is executed automatically within the blockchain, even if it is blocked at some point in VANETs.

7 Performance analysis

In this section, we compare the computational cost, communication overhead, and performance evaluation of our designed scheme with other existing schemes.

We set the security parameters to 80 bits, that is, $\lambda = 80$, the system parameters for the bilinear pairing as "*qbits*=512" and "*rbits*=160", The experimental running hardware configurations are i7-10870H CPU@2.20GHz, 16GB RAM on a HP laptop. Software environment is based on Ubuntu 18.04 OS, using Python version 3.6 and PYPBC version 0.2 for implementation, which uses the PBC library Type A class curves to construct symmetric prime order bilinear groups.

For different signature operations, we tested 100 rounds using the PYPBC library and took their averages as the final results and listed them in Table 2, where T_p denotes a bilinear pairing operation time, T_m denotes a point multiplication operation time, T_e denotes an exponential operation time, T_h denotes a Hash to group operation time.

Table 2 Operation time consumption

Symbol	Definition	Time/ms
T_p	a bilinear pairing operation	1.4481
T_m	a point multiplication operation	0.3526
T_e	an exponential operation	1.1518
T_h	a Hash to group operation	2.4538

7.1 Computational cost

According to the efficiency analysis based on the specific steps in this scheme, the time spent in the vehicle registration phase is $3nT_m + nT_h$, in the key generation phase is $2nT_m + nT_e$, including tracking key generation phase is $nT_m + nT_m$ and vehicle key generation phase is nT_m, in the signature generation phase is $4nT_m + 3nT_e$, in the signature verification phase is $2nT_m$, in the vehicle tracking phase is $nT_m + nT_e + 2nT_p$. According to the time consumption for each operation in Table 2 and the specific steps in our scheme, we calculate the time consumption of the five steps in the scheme with the number of ring members n from 20 to 100, the result is shown in Fig. 6. The computation cost of the schemes [43–46] in key generation, signature generation, signature verification, and tracking processes are computed and the calculation results are listed in Table 3.

Compared with Mao et al.'s scheme [45] in signature generation and verification processes, the computational complexity is almost $\mathcal{O}(n^2)$ when n is the number of ring members. The relationship between computational cost and the number of ring members is not linear. Its computational cost rises sharply as the number of ring members increases. Therefore, we do not compare Mao et al.'s scheme's computational overhead in the signature and verification process with other schemes. Moreover, its scheme is not designed with a malicious user tracking mechanism and does not achieve conditional privacy protection.

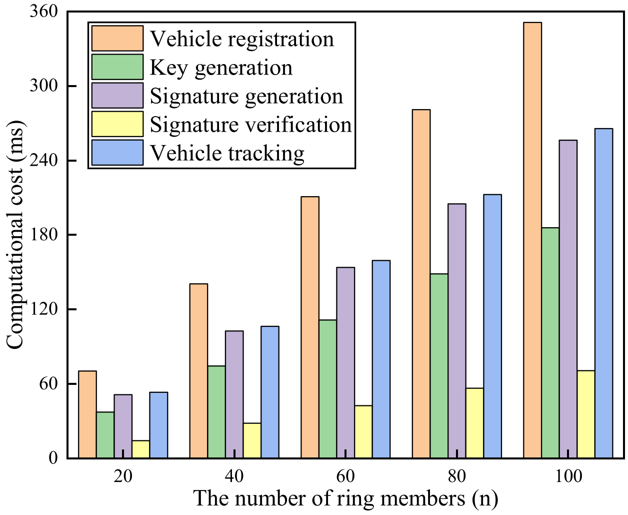


Fig. 6 Computational cost for each step

Table 3 Comparison of computational cost

Scheme	Key generation	Signature generation	Signature verification	Tracking
Ref [43]	$(2n+1)T_e+$ nT_m+2T_h	$(5n+1)Te+$ $(3n-2)T_m+3T_h$	$5nTe+3nT_m$ $+3T_h$	$2nTe+2nT_m$ $+4T_h$
Ref [44]	$3nT_e+4nT_m$	$(3n+1)T_e+nT_m$	$nT_m+(n+1)T_p$	$(2n+1)T_m+$ $(2n+3)T_p$
Ref [45]	nT_m+2T_h	$(4n^2+3n+6)T_m$ $+(n^2+2n)T_h$	$(4n^2+3)T_m$ $+n^2T_h$	*No application*
Ref [46]	$3nTe+nT_m$ $+nT_h$	$(4n+6)Te+$ $(4n-1)T_m$	nT_e+2T_p	$(3n-1)T_m+$ $(2n+2)T_p$
Ours	$2nT_m+nT_e$	$4nT_m+nT_e$	$2nT_m$	nT_m+2nT_p

In key generation process, the Fujisaki and Suzuki's scheme [43] needs n point multiplication operations, two hash to group operations and $2n+1$ exponential operations, and the total time cost is about $nT_m+2T_h+(2n+1)T_e$. We set the number of ring members $n=100$, compared with Fujisaki and Suzuki's scheme [43], Bouakkaz and Semchedine's scheme [44], and Lai et al.'s scheme [46], the key generation consumption of our scheme is reduced by about 31.65%, 61.84%, and 70.34%, respectively. Although, the key generation process cost of our scheme is longer than Mao et al.'s scheme [45]. However, the tracking key and vehicle key of our scheme is generated in advance by the TA and the vehicle during the vehicle registration process, and stored in the OBU of the vehicle, so it does not affect the communication efficiency of VANETs. We set the number of ring members n from 20 to 100, our scheme and other ring signature schemes' computational cost is shown in Fig. 7.

Fig. 8 is a comparison of the computational cost of calculating the signature generation between our scheme and other ring signature schemes [43, 44, 46]. There is a linear relationship between the calculation cost of generating the signature and the number of ring members. Compared to other ring signature schemes, our computational cost is the lowest. When there are 100 ring members, our scheme takes only 256.22 ms to generate ring signatures, which saves the time for signature swapping generation and improves the efficiency of VANETs. In Fujisaki and Suzuki's scheme [43], it requires more point multiplication operations, exponential operations and two additional hash to group operations, resulting in large computational costs, and it takes $(3n-2)T_m+(5n+1)T_e+3T_h$ to generate a ring signature. Compared with our scheme, we set the ring members $n=100$, our computational cost is reduced by 62.84%.

A comparison of the computational cost of signature verification is shown in Fig. 9. It illustrates that as the number of ring members increases, so does the computational cost. The computational consumption of signature verification in the present study is least than that in Fujisaki and Suzuki's scheme [43], Bouakkaz and Semchedine's scheme [44], and Lai et al.'s scheme [46]. In our proposed scheme, the signature verification process is performed by RSUs deployed on both sides of the road, and Edge computing enhances the

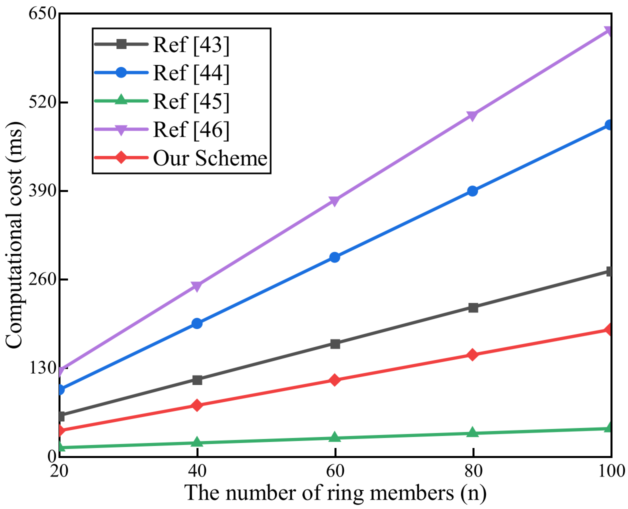


Fig. 7 Computational cost comparison of key generation

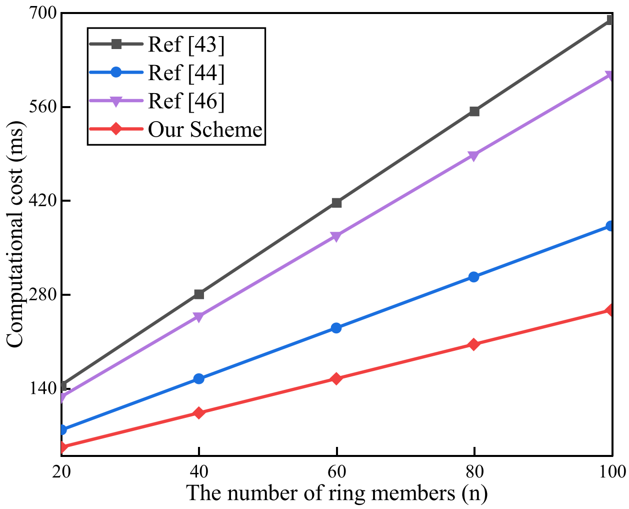


Fig. 8 Computational cost comparison of signature generation

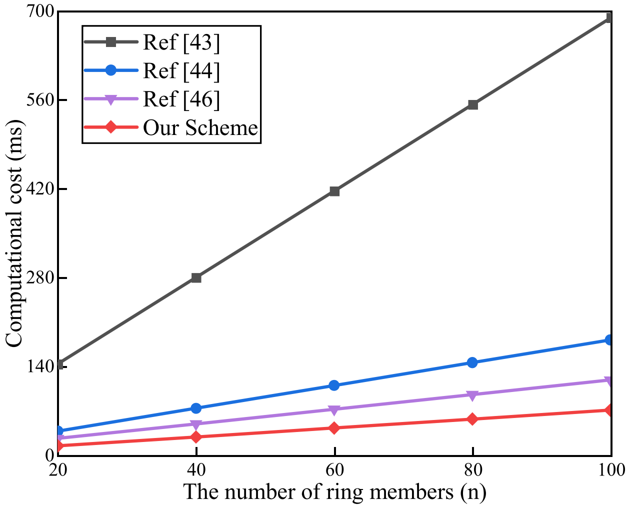


Fig. 9 Computational cost comparison of signature verification

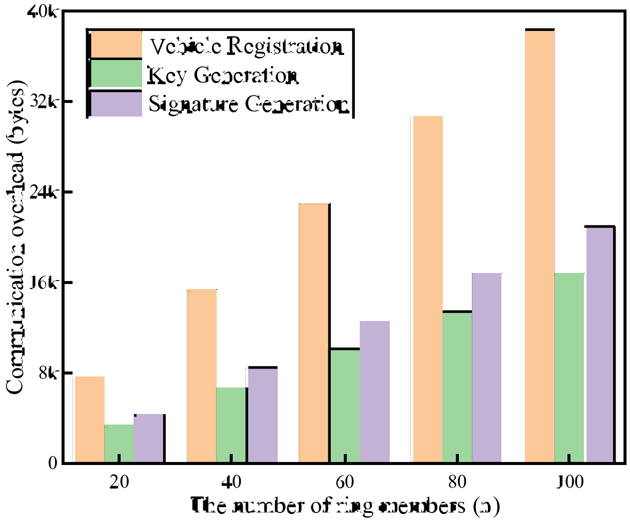


Fig. 11 Communication overhead for each step

computing power of RSUs by accessing them through edge nodes, making signature verification faster. The authenticated messages are broadcasted by RSUs to VANETs and stored in the blockchain, and our scheme effectively improves traffic efficiency.

A comparison of the computational cost of tracking process is shown in Fig. 10. As the number of ring members increases, the time consumed for the tracking process also increases. In our scheme, when there are 100 ring members, we need 324.88 ms to trace the real signer of message m. Compared with Bouakkaz and Semchedine's scheme [44] and Lai et al.'s scheme [46], our scheme has a lower computational overhead. According to Fujisaki and Suzuki's scheme [43], when the number of ring members $n \leq 42$, our scheme has the ability to trace the real signer in a shorter time, when the number of ring members $n > 42$, our scheme has a greater computational overhead in the tracking process.

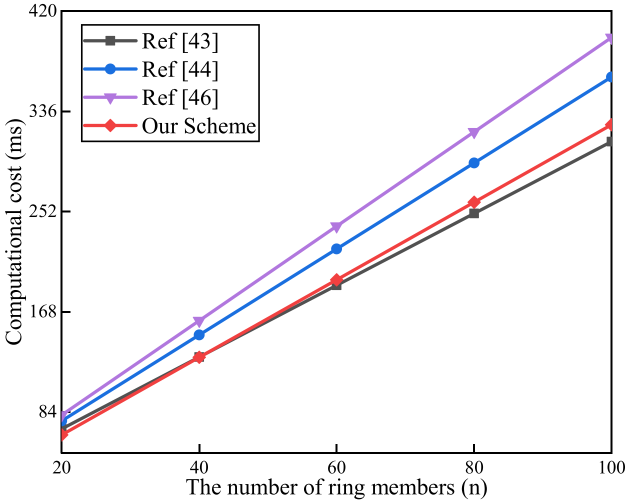


Fig. 10 Computational cost comparison of tracking

7.2 Communication overhead

This section analyses and compares the communication overhead of our scheme and other ring signature schemes [43–46] is shown in Table 4. According to Chen and Chen's paper [7], the size of elements in Z_q^* and G_1 are $20 \times 2 = 40$ bytes and $64 \times 2 = 128$ bytes, respectively. In our scheme, the transmitted parameters in vehicle registration, key generation, and signature generation processes mainly include: vehicle pseudonym $PID_i \in G_1$, tracking private key $x_i \in Z_q^*$, vehicle public key $Y_i \in G_1$, the message's signature T_σ and so on. We set ring members n from 20 to 100, the communication overhead of our scheme is shown in Fig. 11.

In our proposed scheme, the communication overhead is mainly in the signature transmission process, most of the other processes are generated during the vehicle registration process, or in the background operation of the system, and do not affect the efficiency of vehicle data sharing, so we compare the communication overhead of the signature transmission process with Fujisaki änd Suzuki's scheme [43], Bouakkaz and Semchedine's scheme [44], Mao et al.'s

Table 4 Comparison of communication overhead

Scheme	Single ring member (bytes)	N ring members (bytes)
Ref [43]	$4\lvert G_1\rvert + 3\lvert Z_q^*\rvert = 632$	$(3n+1)\lvert G_1\rvert + (2n+1)\lvert Z_q^*\rvert$
Ref [44]	$3\lvert G_1\rvert + \lvert Z_q^*\rvert = 424$	$3n\lvert G_1\rvert + n\lvert Z_q^*\rvert$
Ref [45]	$3\lvert G_1\rvert + 3\lvert Z_q^*\rvert = 504$	$(n+2)\lvert G_1\rvert + (n^2+2)\lvert Z_q^*\rvert$
Ref [46]	$4\lvert G_1\rvert + 2\lvert Z_q^*\rvert = 592$	$(2n+2)\lvert G_1\rvert + (n+1)\lvert Z_q^*\rvert$
Ours	$2\lvert G_1\rvert + 3\lvert Z_q^*\rvert = 376$	$(n+1)\lvert G_1\rvert + (2n+1)\lvert Z_q^*\rvert$

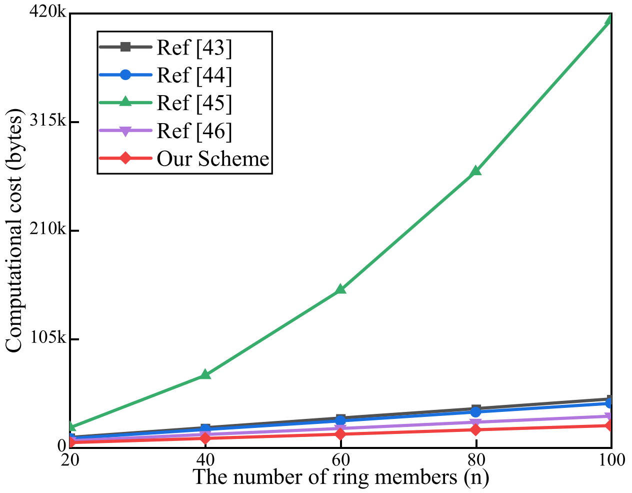


Fig. 12 Communication overhead comparison of signature generation

scheme [45], and Lai et al.'s scheme [46], and the calculation results are listed in Table 4.

Figure 12 compares the communication overhead required by different schemes when transfer of the signature. As ring membership increases, the communication overhead grows. The computational consumption of signature verification in the present study is less than in other schemes [43–46]. The communication complexity of Mao et al.'s scheme [45] is almost $\mathcal{O}(n^2)$, so as the number of ring members increases, the storage and communication overhead required for the ring signature generated by the signature algorithm increases rapidly. As a result, the overall efficiency of the system decreases. When setting the ring members n to 100, the communication overhead of our scheme takes only 20968 bytes, which is a 29.86% reduction in communication overhead compared to Lai et al.'s scheme [46].

7.3 Performance evaluation

In the scheme, we use blockchain as the underlying framework of the system to achieve distributed information storage. The driver's privacy information is stored in TA's secure database, which safeguards the driver's privacy. For drivers who disrupt the normal traffic condition, we deploy smart contracts on the blockchain, which can track these drivers through the signature in the false message. Our scheme introduces vehicle edge computing, which can effectively reduce system latency, process road information faster, and better adapt to the environment where VANETs' network topology changes in real time. Also, edge computing allows data handling locally, which decreases leakage or other security issues, it reduces the system's bandwidth demands, and it enables the system to work even when the network is jammed.

Table 5 compares and analyzes our scheme with other schemes [43–46] in seven aspects: unforgeability, anonymity, traceability, distributed framework, privacy protection, use smart contract and use edge computing. The comparison shows that our solution has better performance and feasibility.

To further illustrate the efficiency of our system operation, we analyze the system overhead. We use Ganache to emulate the Ethernet platform with Node and Web3 as the execution script and write smart contract using the Solidity language. After that, the smart contract are compiled with the truffle framework and deployed in Ethereum. TA's account address is 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4, the address of the smart contract is 0xd9145CCE52D386f254917e481eB44e9943F39138.

The overhead of the blockchain system we designed is mainly based on gas consumed by transactions and smart contracts. Gas is a unit of measurement used to gauge the computational power of executing transactions on the Ethereum platform. All transactions on the Ethernet platform consume a certain amount of gas, and the more complex the task, the more gas is consumed. The cost analysis of the scheme mainly considers the deployment of smart contracts, obtain traceable keys, track signers, and return signers to TA. The dollar consumption in the table refers to the ether price on March 25, 2023, $1eth = 10^{18}wei = 1744dollars$.

The REMIX IDE and Solidity version 0.8.0 with a specification of Ubuntu 18.04, 16 GB RAM, Intel Core i7-10870H CPU@2.20GHz, 64-bit OS are used for experimental validation. Each execution requires a gas price cap and a unit price per gas. The miner determines the gas price

Table 5 Functional comparison of different schemes

Performance	Ref [43]	Ref [44]	Ref [45]	Ref [46]	Our scheme
Unforgeability	✗	✓	✓	✓	✓
Anonymity	✓	✓	✓	✓	✓
Traceability	✓	✓	✗	✓	✓
Distributed Framework	✗	✗	✗	✓	✓
Privacy Protection	✗	✓	✓	✗	✓
Use Smart Contract	✗	✗	✗	✗	✓
Use Edge Computing	✗	✗	✗	✗	✓

Table 6 Cost consumption

Process execution	Gas used	Actual cost (wei)	USD (dollars)
Deploy smart contract	1195397	1039475	1.8128e-9
Obtain traceable keys	91887	79901	1.3935e-10
Tracking signer	241114	209664	3.6565e-10
Return signer to TA	72180	62765	1.0946e-10

and if the execution operation gas price is lower than the price determined by the miner, the miner will refuse to perform this operation. The costs measured by the experiment are shown in Table 6. As described in the table, the smart contract deployment is only performed once and the cost is $1.8128e-9. The costs of *Obtain traceable keys*, *Tracking signer*, and *Return signer to TA* operation is $1.3935e-10, $3.6565e-10, and $1.0946e-10, respectively.

8 Conclusion

In this paper, we have proposed a traffic data security sharing scheme based on blockchain and traceable ring signature for VANETs. The scheme uses smart contract to track the illegal vehicles and return the results to trusted authority (TA). The TA holds the source of the signature message and determines the penalties. This facilitates safeguarding the privacy of drivers and improving the security of VANETs with sharing messages. Edge computing is utilized to solve the problem of low computing power of VANETs. Vehicle registration and signer tracking can be offloaded through edge nodes to cloud servers with greater computer storage capacity, they can also improve the efficiency of VANETs. The security analysis shows that the proposed traceable ring signature algorithm has strong security and anonymity. Compared with other schemes, our scheme has better features such as signer tracking, use of smart contract, etc. Performance analysis shows that our proposed scheme has smaller computational cost and communication overhead.

Author contributions Writing-original draft preparation, Jiaming Lai; writing-review and project design, Xiaohong Zhang, Jiaming Lai; polish and proofread, Ata Jahangir Moshayedi. All authors have read and agree to the published version of the manuscript.

Funding This work is jointly supported by the National Natural Science Foundation of China (Nos. 61763017, 51665019), Key Program of Natural Science Foundation of Jiangxi Province (Grant No. 20212ACB202003), the Innovation Fund for Graduate Students in Jiangxi Province (Grant No: YC2022-S679).

Data availibility Data used to support the study already included in the manuscript.

Declarations

Ethics approval This article does not contain any studies with human participants or animals performed by any of the authors.

Consent to publish All authors have read and agreed to the published the manuscript.

Conflicts of interests The authors declare no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Dey KC, Mishra A, Chowdhury M (2014) Potential of intelligent transportation systems in mitigating adverse weather impacts on road mobility: A review. IEEE Trans Intell Transp Syst 16(3):1107–1119
2. Al-shareeda MA, Alazzawi MA, Anbar M, Manickam S, Al-Ani AK (2021) A Comprehensive Survey on Vehicular Ad Hoc Networks (VANETs). In 2021 International Conference on Advanced Computer Applications (ACA), IEEE, pp 156-160 https://doi.org/10.1109/ACA52198.2021.9626779
3. Jiang H, Hua L, Wahab L (2021) SAES: A self-checking authentication scheme with higher efficiency and security for VANET. Peer-to-Peer Network Appl 14(2):528–540
4. Azees M, Vijayakumar P, Deboarh LJ (2017) EAAP: Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks. IEEE Trans Intell Transp Syst 18(9):2467–2476
5. Kenney JB (2011) Dedicated short-range communications (DSRC) standards in the United States. Proc IEEE 99(7):1162–1182
6. Manivannan D, Moni SS, Zeadally S (2020) Secure authentication and privacy-preserving techniques in Vehicular Ad-hoc NETworks (VANETs). Veh Commun 25:100247
7. Chen Y, Chen J (2021) CPP-CLAS: efficient and conditional privacy-preserving certificateless aggregate signature scheme for VANETs. IEEE Internet Things J 9(12):10354–10365
8. Wu Y, Dai H N, Wang H (2020) Convergence of blockchain and edge computing for secure and scalable IIoT critical infrastructures in industry 4.0. IEEE Internet Things J 8(4):2300-2317
9. Ayaz F, Sheng Z, Tian D, Guan YL (2020) A proof-of-quality-factor (PoQF)-based blockchain and edge computing for vehicular message dissemination. IEEE Internet Things J 8(4):2468–2482
10. Li X, Liu T, Obaidat MS, Wu F, Vijayakumar P, Kumar N (2020) A lightweight privacy-preserving authentication protocol for VANETs. IEEE Syst J 14(3):3547–3557
11. Zeng M, Xu H (2019) Mix-context-based pseudonym changing privacy preserving authentication in VANETs. Mobile Information Systems 2019:1–9

12. Zhong H, Han S, Cui J, Zhang J, Xu Y (2019) Privacy-preserving authentication scheme with full aggregation in VANET. Information Sciences 476:211–221
13. Zheng Z, Xie S, Dai H, Chen X, Wang H (2017) An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In 2017 IEEE international congress on big data (Big Data congress), IEEE, pp 557-564. https://doi.org/10.1109/BigDataCongress.2017.85
14. Upadhyay A, Mukhuty S, Kumar V, Kazancoglu Y (2021) Blockchain technology and the circular economy: Implications for sustainability and social responsibility. J Cleaner Prod 293:126130
15. Nguyen DC, Ding M, Pham QV et al (2021) Federated learning meets blockchain in edge computing: Opportunities and challenges. IEEE Internet Things J 8(16):12806–12825
16. Zhang D, Yu FR, Yang R (2021) Blockchain-based multi-access edge computing for future vehicular networks: A deep compressed neural network approach. IEEE Trans Intell Transp Syst 23(8):12161–12175
17. Poongodi M, Bourouis S, Ahmed AN et al (2022) A novel secured multi-access edge computing based vanet with neuro fuzzy systems based blockchain framework. Comput Commun 192:48–56
18. Gao J, Agyekum KOBO, Sifah EB et al (2019) A blockchain-SDN-enabled Internet of vehicles environment for fog computing and 5G networks. IEEE Internet Things J 7(5):4278–4291
19. Fang W, Chen W, Zhang W, Pei J, Gao W (2020) Wang G (2020) Digital signature scheme for information non-repudiation in blockchain: a state of the art review. EURASIP Journal on Wireless Communications and Networking 1:1–15
20. Hubaux JP, Capkun S, Luo J (2004) The security and privacy of smart vehicles. IEEE Security and Privacy 2(3):49–55
21. Raya M, Hubaux JP (2007) Securing vehicular ad hoc networks. J Comput Secur 15(1):39–68
22. Wasef A, Lu R, Lin X, Shen X (2010) Complementing public key infrastructure to secure vehicular ad hoc networks [security and privacy in emerging wireless networks]. IEEE Wireless Commun 17(5):22–28
23. Benarous L, Kadri B, Bouridane A (2020) Blockchain-based privacy-aware pseudonym management framework for vehicular networks. Arabian Journal for Science and Engineering 45(8):6033–6049
24. Shamir A (1984) Identity-based cryptosystems and signature schemes. In Workshop on the theory and application of cryptographic techniques, Springer, Berlin, Heidelberg, pp 47-53. https://doi.org/10.1007/3-540-39568-7_5
25. Deng L, Shao J, Hu Z (2021) Identity based two-party authenticated key agreement scheme for vehicular ad hoc networks. Peer-to-Peer Network Appl 14(4):2236–2247
26. Akram W, Mahmood K, Li X, Sadiq M, Lv Z, Chaudhry SA (2022) An energy-efficient and secure identity based RFID authentication scheme for vehicular cloud computing. Comput Networks 217:109335
27. Lei A, Cruickshank H, Cao Y, Asuquo P, Ogah CP, Sun Z (2017) Blockchain-based dynamic key management for heterogeneous intelligent transportation systems. IEEE Internet of Things J 4(6):1832–1843
28. Ma Z, Zhang J, Guo Y, Liu Y, Liu X, He W (2020) An efficient decentralized key management mechanism for VANET with blockchain. IEEE Trans on Veh Technol 69(6):5836–5849
29. Malhi AK, Batra S, Pannu HS (2019) An efficient privacy preserving authentication scheme for vehicular communications. Wireless Personal Communications 106(2):487–503
30. Rasheed AA, Mahapatra RN, Hamza-Lup FG (2019) Adaptive group-based zero knowledge proof-authentication protocol in vehicular ad hoc networks. IEEE Trans Intell Transp Syst 21(2):867–881
31. Li K, Lau WF, Au MH, Ho IW, Wang Y (2020) Efficient message authentication with revocation transparency using blockchain for vehicular networks. Comput and Electr Eng 86:106721
32. Coruh U, Bayat O (2022) ESAR: Enhanced Secure Authentication and Revocation Scheme for Vehicular Ad Hoc Networks. Journal of Information Security and Applications 64:103081
33. Wang S, Mao K, Zhan F, Liu D (2020) Hybrid conditional privacy-preserving authentication scheme for VANETs. Peer-to-Peer Network Appl 13(5):1600–1615
34. Li H, Pei L, Liao D, Sun G, Xu D (2019) Blockchain meets VANET: An architecture for identity and location privacy protection in VANET. Peer-to-Peer Network Appl 12(5):1178–1193
35. Shrestha R, Bajracharya R, Shrestha AP, Nam SY (2020) A new type of blockchain for secure message exchange in VANET. Digital Commun Networks 6(2):177–186
36. Feng Q, He D, Zeadally S, Liang K (2019) BPAS: Blockchain-assisted privacy-preserving authentication system for vehicular ad hoc networks. IEEE Trans Ind Inf 16(6):4146–4155
37. Lin C, He D, Huang X, Kumar N, Choo KR (2020) BCPPA: A blockchain-based conditional privacy-preserving authentication protocol for vehicular ad hoc networks. IEEE Trans Intell Transp Syst 22(12):7408–7420
38. Gong J, Mei Y, Xiang F, Hong H, Sun Y, Sun Z (2021) A data privacy protection scheme for Internet of things based on blockchain. Trans Emerging Telecommun Technol 32(5):e4010
39. Rivest RL, Shamir A, Tauman Y (2001) How to leak a secret. In International conference on the theory and application of cryptology and information security, Springer, Berlin, Heidelberg, pp 552-565. https://doi.org/10.1007/3-540-45682-1_32
40. Wang L, Zhang G, Ma C (2008) A survey of ring signature. Front Electr Electron Eng China 3(1):10–19
41. Ma C, Zhu J, Liu M, Zhao H, Liu N, Zou X (2021) Parking edge computing: parked-vehicle-assisted task offloading for urban VANETs. IEEE Internet Things J 8(11):9344–9358
42. Bender A, Katz J, Morselli R (2009) Ring signatures: Stronger definitions, and constructions without random oracles. Journal of Cryptology 22(1):60–79
43. Fujisaki E, Suzuki K (2008) Traceable ring signature. IEICE Trans Fundam Electron Commun Comput Sci 91(1):83–93
44. Bouakkaz S, Semchedine F (2020) A certificateless ring signature scheme with batch verification for applications in VANET. Journal of Information Security and Applications 55:102669
45. Mao X, You L, Cao C, Cao C, Hu G, Hu L (2021) Linkable Ring Signature Scheme Using Biometric Cryptosystem and NIZK and Its Application. Secur Commun Netw 2021
46. Lai C, Ma Z, Guo R, Zheng D (2022) Secure medical data sharing scheme based on traceable ring signature and blockchain. Peer-to-Peer Network Appl 15(3):1562–1576

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Xiaohong Zhang received the B.S. degree in physics from Jiangxi Normal University, Jiangxi, China, in 1984. The M.S. degree in Optical Information Processing from Chinese Academy of Sciences, Changchun, China, in 1990, and the PH.D degree in control theory, information safety, from the University of Science and Technology Beijing (USTB) and Beijing University of Posts and Telecommunications (BUPT) in 2002, 2006, respectively. She was a Visiting Scholar with the University of California, Berkeley, USA, from 2014 to 2015. She is currently a full Professor with the Department of College of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, China. Her main research interests are blockchain technology, information security, nonlinear dynamics, wireless sensor network, etc.

Jiaming Lai received the B.S. degree in software engineering from Jiangxi University of Science and Technology, Jiangxi, China. In 2021, he is currently pursuing the M.S. degree in electronic information from Jiangxi University of Science and Technology, Jiangxi, China. His current research includes blockchain technology and intelligent transportation system.

Ata Jahangir Moshayedi received the B.S. degree in Power electrical Engineering from Azad University, Iran in 2004, The M.S. degree in Instrumentation Science from Savitribai Phule Pune University (formerly named Pune University), India, in 2009 and PhD in Electronic Science in electronic and Robotic from Savitribai Phule Pune University (formerly named Pune University), India in 2015. Currently working as Associate professor at College of Information Engineering, Jiangxi University of Science and Technology, China. IEEE member, Instrument Society of India as a Life Member, Lifetime Member of Speed Society of India, member of the editorial team of various conferences and journals like; International Journal of Robotics and Control, JSME, Bulletin of Electrical Engineering and Informatics, International Journal of Physics and Robotics Applied Electronics, etc. His research interest includes: Robotics and Automation, Artificial Intelligence.