Article | Received 20 December 2025; Revised 11 February 2026; Accepted 26 February 2026; Published 25 March 2026
https://doi.org/10.55092/blockchain20260003

An ethereum-based fully distributed authentication mechanism in VANETs

Chunyan Liu[1,2], Hongkun Tian[3], Tao Feng[4], Fuliang Lin[3] and Xiaoqin Feng[3,*]

[1] School of Economics and Management, Lanzhou University of Technology, Lanzhou, China
[2] School of Economics and Management, Southest University, Nanjing, China
[3] School of Information Science & Engineering, Lanzhou University, Lanzhou, China
[4] School of Computer and Communication, Lanzhou University of Technology, Lanzhou, China

* Correspondence author; E-mail: fxq@lzu.edu.cn.

Highlights:

- Certificate-free vehicle identity authentication achieved via Ethereum-based graph of trust.
- Decentralized and auditable management of vehicle identities and pseudonyms using smart contracts.
- Prototype evaluation demonstrates reduced authentication latency and low resource consumption.

Abstract: Secure and efficient identity authentication is a fundamental requirement in vehicular ad-hoc networks (VANETs); however, it remains challenging due to the highly dynamic network topology, stringent latency constraints, and the need for conditional privacy preservation. Existing authentication schemes either rely on public key infrastructures (PKI) with complex certificate management or introduce partially decentralized designs that still depend on trusted authorities, leading to inefficiencies and single points of failure. In this paper, we propose EBDA, an Ethereum-based fully distributed authentication mechanism for VANETs. The core innovation of EBDA is to replace the traditional PKI certificate system with a blockchain-maintained Graph of Trust (GoT). Through three dedicated smart contracts, EBDA fully decentralizes the management of vehicle identities and pseudonyms. Vehicles use pseudonyms to preserve privacy in Vehicle-to-Vehicle communications, while authentication is achieved certificate-free via transitive trust within the GoT. Importantly, latency-sensitive operations like message verification are executed off-chain through local checks, meeting VANETs' strict real-time requirements. A prototype implementation and extensive evaluations demonstrate that EBDA significantly reduces authentication latency by at least 22.93% compared with representative blockchain-assisted and PKI-based baselines while maintaining low computational and storage overhead. These results confirm the feasibility of deploying GoT-based decentralized authentication in practical VANET environments.

Keywords: certificate-free authentication; identity-claim-based graph of trust; decentralized identity management; off-chain verification; VANETs

1. Introduction

Identity authentication in vehicular ad-hoc networks (VANETs) faces unique challenges that distinguish it from conventional mobile or IoT networks. Vehicles operate in highly dynamic topologies with frequent topology changes, stringent real-time constraints, and large-scale participation. At the same time, authentication mechanisms must simultaneously ensure low latency, conditional privacy preservation, and accountability, making traditional security solutions difficult to apply directly.

Although centralized public key infrastructures (PKI)-based methods have been widely adopted, they suffer from inherent limitations such as single points of failure, complex certificate lifecycle management, and high communication overhead. To address these issues, recent studies have explored decentralized or blockchain-assisted authentication schemes; however, many still rely on trusted authorities for registration, key management, or revocation, or incur substantial computational overhead due to complex cryptographic operations or mining requirements. As summarized in Table 1, most representative schemes remain PKI-dependent, introduce potential single points of failure, or impose high resource consumption on vehicles. These limitations indicate that existing solutions do not yet fully satisfy the stringent requirements of VANET identity authentication, motivating the need for a truly certificate-free and lightweight decentralized framework.

Table 1. Feature comparison.

Schemes	Decentralization	PKI-based	Single point of failure	Accountability	High resource consumption
[1]	Yes	Yes	Yes	Yes	Yes
[2]	No	Yes	Yes	Yes	Yes
[3]	No	Yes	Yes	Yes	No
[4]	No	No	Yes	Yes	No
[5]	No	Yes	Yes	Yes	Yes
[6]	No	No	Yes	Yes	No
[7]	Yes	Yes	Yes	Yes	No
[8]	Yes	No	Yes	Yes	Yes
[9]	Yes	Yes	Yes	Yes	No
[10]	Yes	No	No	Yes	No
[11]	Yes	No	No	Yes	No
[12]	Yes	Yes	Yes	Yes	No

Currently, researchers are exploring new schemes that combine blockchain technology with vehicle identity authentication in VANETs. Some schemes do not rely on certificates. For example, Blockchain-Assisted Certificateless Key Agreement Protocol [13] achieved decentralized vehicle identity authentication through a key agreement protocol based on elliptic curves. Nevertheless, it faces performance bottlenecks in high-density traffic due to multiple elliptic curve point multiplications. Blockchain-Assisted Privacy-Preserving Authentication System (BPAS) [14] used blockchain and smart contracts to store and query vehicle public keys for vehicle-to-vehicle (V2V) identity authentication, achieving decentralized and efficient authentication. However, it introduced a trusted authority (TA) for vehicle registration, risking a single point of failure. The other researches rely on PKI, where resource-rich Certificate Authorities (CAs) handle most of

the computational and storage-intensive tasks. For example, DrivMan [15] proposed a decentralized system where roadside units (RSUs) act as CAs and blockchain nodes, issuing certificates and using smart contracts for data authentication. However, vehicles faced high costs for storing and querying the certificate revocation list (CRL). Blockchain-based Public Key Infrastructure [1] used a blockchain-based pseudonym management method where CAs issue long-term certificates, and the blockchain stores and revokes vehicle pseudonyms. While it achieves efficient authentication, the requirement for vehicles to act as miners significantly increases their resource demands. Besides, these solutions still rely on CA, leaving them with a possible single point of failure.

To solve the aforementioned issues, we propose an Ethereum-based fully distributed authentication mechanism (EBDA) for VANETs. A graph of trust (GoT) is constructed, which is implemented through smart contracts deployed on Ethereum for collaborative maintenance. By replacing PKI certificates with GoT for identity verification, efficiency is significantly improved. Unlike existing schemes, EBDA eliminates multi-tiered CAs, avoiding single points of failure. It manages the entire lifecycle of vehicle pseudonyms, from registration to revocation, on the blockchain, ensuring low resource demands on vehicles during pseudonym verification.

It is important to distinguish the proposed certificate-free authentication from existing certificateless schemes. Certificateless approaches typically eliminate explicit certificates but still rely on trusted authorities for key generation, registration, or partial private key issuance, thereby retaining centralized trust dependencies. In contrast, EBDA removes both certificates and centralized trust anchors by encoding trust relationships directly into a GoT, which is collaboratively maintained on the blockchain through smart contracts. Moreover, the reported authentication latency reduction of at least 22.93% is measured against BPAS and Traceable Blockchain-based Access Authentication (TBAA), two representative blockchain-assisted VANET authentication schemes, under identical simulation settings such as vehicle density, communication model, and cryptographic parameters.

Specifically, we make the following contributions:

- Certificate-free identity authentication based on GoT. GoT reflects the trust relationships between vehicle owners and other entities, allowing VANET members to establish trust with vehicles by simply querying the GoT. By adding the identity claims of registered vehicles to GoT, the subordinate relationships with their owners can be established. Thus, a trust path between the verifier and the vehicle can be simply traced through their identity claims, which replaces the expensive certificate.
- A decentralized and auditable vehicle identity and pseudonym management scheme. We design three smart contracts (identity management contract, pseudonym registration contract, and pseudonym revocation contract) deployed on three separate blockchains to automate vehicle identity and pseudonym management. By integrating GoT, the scheme enables decentralized identity management and authentication with accountable governance.
- Full implementation of the designed prototype. We simulate the comprehensive prototype system. Besides, a systematic performance is evaluated and compared to highlight the efficiency of our design.

2. Related work

In this section we review the most relevant works in existing literature. Efficient vehicle identity verification is crucial for the practical implementation of VANETs. PKI, where a CA issues temporary pseudonym certificates, remains the most typical authentication scheme. Additionally, cryptographic primitives are employed to secure authentication. Recently, blockchain technology has emerged as a key focus for vehicle identity authentication in VANETs.

Ensuring secure and efficient authentication in VANETs has been a major concern for researchers. Shuo *et al.* [2] proposed the APKI scheme, combining PKI with timestamp signatures for effective anonymous authentication, but it demands high computational and storage resources due to key and certificate management. Wang *et al.* [3] introduced the Two-Factor Lightweight Integrated Privacy-Preserving Authentication scheme, which uses decentralized CA and biometric-based two-factor authentication for partial decentralization, though stolen biometric data presents significant security risks. Li *et al.* [4] proposed a lightweight authentication protocol using hash functions and Exclusive OR operations for efficient data transmission, but it lacks conditional privacy protection. Calandriello *et al.* [5] developed an efficient and robust pseudonym-based authentication mechanism with group signatures. However, the large overhead of CRL storage and queries remains an issue. Zhong *et al.* [6] combined aggregate signatures and pseudonym techniques, reducing communication and storage overhead. Nevertheless, the complexity of aggregate signatures and dependence on trusted authorities present practical challenges.

Blockchain 3.0 has expanded from finance to sectors like government, healthcare, and transportation. Recently, researchers have applied blockchain to VANETs to improve identity authentication with its decentralized, traceable, and tamper-proof features. Zheng *et al.* [7] proposed a traceable decentralized system by storing authentication records on the blockchain. Dwivedi *et al.* [8] designed a decentralized batch authentication protocol using elliptic curve cryptography. Lu *et al.* [9] recorded TA activities on blockchain and used the Merkle Patricia Tree to replace traditional CRLs. Akhter *et al.* [10] introduced a protocol that prioritizes urgent authentication tasks based on context. Li *et al.* [11] recorded safety beacon message (SBMs) hashes on blockchain, protecting vehicle privacy with multiple sub-identities. Feng *et al.* [12] proposed a framework that stores public keys on blockchain instead of using certificates. Moussaoui *et al.* [1] developed a distributed pseudonym management architecture with two blockchains for registration and revocation, but it requires vehicles to act as miners, increasing their computational resource demands.

In summary, existing literature focuses on enhancing the efficiency and security of identity authentication in VANETs. Despite employing various technologies like PKI, cryptographic primitives, and blockchain, these schemes face challenges including high computational resource demands, single points of failure, and solution complexity. The features of the schemes we have discussed are shown in Table 1.

3. Problems definitions

In this section, we describe the problem we tackle and outline the existing optimizations.

3.1. Challenges in identity management

Although the PKI scheme is amongst the most popular method adopted for vehicle authentication in VANETs, it has some major limitations:

- Current PKI-based scheme rely heavily on a central CA to issue and manage certificates. The centralized approach presents a single point of failure, making the system vulnerable to adversary attacks.
- Authenticating vehicle identities through public key certificates cannot meet security requirements such as protecting location privacy and unlinkability.
- Managing and distributing certificates in dynamic networks incurs significant overhead. The need to frequently update and distribute certificates to ensure security and privacy increases the complexity and operational costs.

3.2. Smart contracts in VANETs

A smart contract is a self-executing program that runs on a blockchain platform, such as Ethereum, to automate processes according to predefined terms of agreement. Ethereum is a decentralized and secure data ledger where transactions and smart contract executions are recorded in an append-only chain of blocks. This structure ensures transparency, immutability, and tamper resistance, as all changes are validated and permanently stored across the participating nodes in the network. A smart contract functions as a digital agreement, with terms encoded directly into the program. Once deployed on Ethereum, the contract automatically enforces its terms without requiring intermediaries or trusted authorities. This reduces arbitration costs, minimizes fraud risks, and ensures reliable operation in a decentralized manner.

Previous research has highlighted the potential of smart contracts in VANETs for various applications. For instance, Wei *et al.* [16] used smart contracts for vehicle access control. In [17], they ensured the non-repudiation of task execution information in autonomous vehicular clouds using smart contracts. Additionally, Javaid *et al.* [15] integrated blockchain and smart contracts for trust management in data sharing, enhancing data integrity with physical unclonable functions (PUFs). These studies demonstrate how smart contracts can decentralize control, automate processes, reduce reliance on centralized entities, and improve system efficiency.

While blockchain has been widely recognized as a suitable infrastructure for decentralized identity management, its effectiveness depends critically on how trust relationships are represented and updated. Ethereum smart contracts provide programmable, autonomous, and verifiable execution environments that are particularly well suited for dynamic trust management in VANETs, where trust relationships evolve over time and must be enforced without centralized arbitration. Unlike decentralized identifier (DID)-based solutions that primarily focus on identifier resolution and credential presentation, the proposed GoT explicitly models trust semantics among entities and devices. GoT enables identity verification through transitive trust relationships rather than credential exchange, which better aligns with the highly dynamic and proximity-driven interactions in VANETs. Therefore, GoT is adopted as a more suitable trust abstraction for certificate-free authentication in vehicular networks.

3.3. Security model

The security and privacy requirements are defined as follows:

- Single registration: The vehicles need to register only once before joining VANETs for communication.
- Message authentication: Due to the open and insecure nature of communication in VANETs, the message receiver needs to verify the legitimacy of the sender's identity and the integrity of the corresponding message.
- Conditional privacy preservation: The real identity of each vehicle should be invisible to RSUs or other vehicles, and no adversary should be able to deduce the sender's real identity from the messages. However, authorities should have the capability to obtain and reveal the real identity of malicious vehicles.
- Resistance to replay attack: To determine whether to retain the message, the receiver should verify the freshness of a message before processing it, thereby avoiding replay attacks.
- Resistance to man-in-the-middle attack: Any adversary should not be able to forge or tamper with the messages.
- Resistance to denial-of-service attack: The proposed scheme should be capable of withstanding denial-of-service attacks, ensuring that such attacks do not disrupt or compromise the normal operations of the system.

3.4. Threat model

We define the threat model as follows. There are two types of adversaries considered, namely vehicles and external adversaries other than vehicles. The adversaries have the following malicious behaviors. (1) Eavesdropping on vehicle communication channels to collect traffic information. (2) Attempting to decrypt ciphertexts to obtain the vehicle privacy. (3) Modifying a portion of an intercepted valid message and forward the modified message to the target recipient. (4) Fabricating a malicious message by impersonating a valid user. We assume that TAs and RSUs are honest, so the real identity of each vehicle will not be leaked. Same to general public blockchain systems, we assume that a 51% attack is unlikely to occur.

4. EBDA: detailed design

In this section, we provide a full description of EBDA including system model, smart contract function design, and normal-case operation.

4.1. System model

The proposed scheme comprises five main components. It includes vehicles, RSUs, the identity management blockchain (IMBC), the pseudonym registration blockchain (PBC), and the pseudonym revocation blockchain (PRBC), as shown in Figure 1. We define the required notations in Table 2.

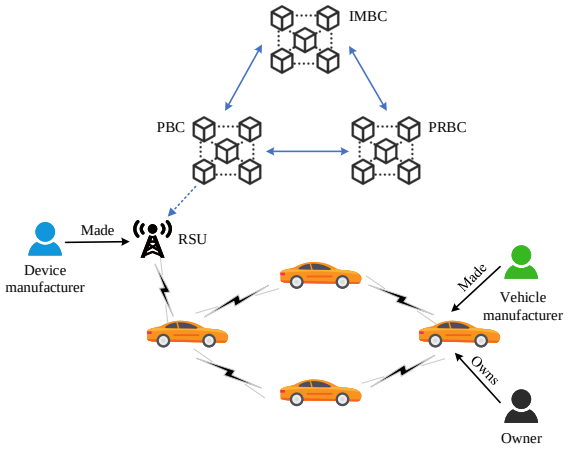


Figure 1. System model of EBDA.

Table 2. Notations.

Notations	Descriptions
(pk, sk)	Vehicle public and private keys
id	Entity or vehicle identity claim
lp	License plate number of vehicle
v	Vin code of the vehicle
$\mathscr{A}$	Ethereum account address
p	Vehicle pseudonym
$H()$	Hash function
$\langle x \rangle_{sk}$	Signature of x with private key sk
$Vrfy()$	Verify signature function
$M(x) \Rightarrow y$	Matching function

It is important to clarify that EBDA does not rely on real-time blockchain interactions for V2V or Vehicle-to-Infrastructure (V2I) message authentication. Blockchain operations are confined to low-frequency control-plane procedures, such as vehicle registration, pseudonym issuance, and revocation. Time-critical message authentication is performed entirely off-chain through local pseudonym verification and signature checking, which meets the millisecond-level latency requirements of VANETs.

- Vehicle: The vehicular ad-hoc network formed by vehicles is one of the main components of our proposed scheme. Vehicles can communicate with other entities using on-board units (OBUs).
- RSU: RSUs have ample storage and computing resources, acting as full nodes of the blockchain. They handle blockchain transactions, store transaction information, and provide interfaces to access data on the blockchain and invoke smart contract functions.
- IMBC: The IMBC is an Ethereum blockchain that provides vehicle identity registration and verification services. It uses the smart contract to register vehicle identities and construct the GoT.
- PBC: The PBC is an Ethereum blockchain used for registering and verifying vehicle pseudonyms. Vehicles can register pseudonyms and verify the validity of others' pseudonyms through the smart contract on the PBC.
- PRBC: The PRBC is used for revoking and storing invalid vehicle pseudonyms in our proposed scheme. Authorities can query the initial pseudonym of a malicious vehicle through the PRBC.

4.2. Smart contract function design

This section presents the detailed design of smart contracts deployed on the three Ethereum blockchains (IMBC, PBC, and PRBC) in EBDA, as shown in Algorithms 1–3 (Contract 1, Contract 2, and Contract 3). These three smart contracts respectively implement the functionalities of vehicle identity registration, vehicle pseudonym registration, and vehicle pseudonym revocation.

Algorithm 1 Contract 1—Identity management

persistent variables:
  maintained by vehicles:
    $id, v, lp, pk, sk, \mathscr{A}_o, id_o$ ▷ $\mathscr{A}_o$ owner address, id_o owner identity claim.
  maintained by IMBC only:
    id_m, id_d ▷ identity claims of vehicle manufacturers, device manufacturers.
    $\mathscr{T}$ ▷ transaction timestamp.
function $Register(id, v, lp, id_o, \mathscr{A}_o, \langle x \rangle_{sk})$
  ▷ signature object: $x \leftarrow \{id, v, lp, id_o, \mathscr{A}_o\}$
  if $Vrfy(\langle x \rangle_{sk}) = 0$; **return** "invalid signature"
  else $p_i \leftarrow H(id, \mathscr{T})$; ▷ generate an initial pseudonym for the vehicle.
    $M_{info}(p_i) \Rightarrow \{id, lp, v, \mathscr{A}_o\}$;
    $register\_p_i(p_i, id) =$ "successfully registered";
  ▷ register the initial pseudonym.
    $GoTaddnode(id, id_o)$;
return "vehicle successfully registered"
end function

function $constructGoT(id_{ent1}, id_{ent2})$
  ▷ used by IMBC only.
  $M_{trust}(id_{ent1}) \Rightarrow id_{ent2}$; ▷ the entity 1 trusts the entity 2.
return "successfully constructed"
end function

function $queryInfo(p_i)$ ▷ used by authorities only.
  if $M_{info}(p_i) = null$;
    return "vehicle information not found"
  else $\{id, lp, v, \mathscr{A}_o\} \leftarrow M_{info}(p_i)$;
    return $\{id, lp, v, \mathscr{A}_o\}$
end function

function $GoTaddnode(id_{dev}, id_{ent})$ ▷ match the device with the corresponding entity.
  $M_{belong}(id_{dev}) \Rightarrow id_{ent}$;
end function

function $modifyGoT(id)$ ▷ used by PRBC only.
  if $M_{belong}(id) = null$; **return** "invalid id"
  else $M_{belong}(id) = null$;
return "successfully modified"
end function

function $isTrustworthy(id_{dev1}, id_{dev2})$
  ▷ Used to build trust relationships between two devices.
  if $M_{belong}(id_{dev1}) \neq null \ \& \ M_{belong}(id_{dev2}) \neq null$;
    $id_{ent1} \leftarrow M_{belong}(id_{dev1})$;
    $id_{ent2} \leftarrow M_{belong}(id_{dev2})$;
    if $M_{trust}(id_{ent1}) = id_{ent2}$; ▷ verify whether the entity associated with device 1 trusts the entity associated with device 2.
      return "trustworthy"
    else return "untrustworthy"
  else return "invalid id"
end function

The Contract 1 is deployed on the IMBC and responsible for registering vehicles and generating the GoT. The functions in the contract are explained as follows:

- Register: This function takes six parameters id, v, lp, id_o, $\mathscr{A}_o$, $\langle x \rangle_{sk}$. After verifying the validity of the vehicle's signature (line 8), it generates the initial pseudonym for the vehicle (line 9), stores the vehicle information (line 10), registers the initial pseudonym (line 11), and adds the vehicle's identity claim as a new node in the trust graph (line 12). Finally, it returns the "vehicle successfully registered".
- constructGoT: This function takes two parameters id_{ent1} and id_{ent2}. It is used to construct GoT by matching the trust relationships between two entities (line 17). This function is automatically invoked by IMBC during system initialization based on the trust relationship between entities.
- queryInfo: This function takes the parameter p_i, and queries the real information of the vehicle using p_i as the index (line 24), then returns the id, lp, v, $\mathscr{A}_o$ of the vehicle.
- GoTaddnode: This function takes two parameters id_{dev} and id_{ent}. It maps the identity claim of the device and the corresponding entity to indicate a subordinate relationship (line 29).
- modifyGoT: This function takes the parameter id. It removes the subordinate relationship between the device and the corresponding entity by disabling the record in M_{belong} (line 34).
- isTrustworthy: This function takes two parameters id_{dev1} and id_{dev2}, and verifies two conditions. In one condition, ids of the two devices have both been added to GoT (line 39). In another condition a trust relationship exists in GoT between the entities associated with the devices (line 42). If both conditions are successfully verified, the function returns "trustworthy" indicating that the trust relationship has been established successfully, otherwise it returns "untrustworthy" or "invalid id".

The PseudonymRegistration contract is deployed on the PBC and is responsible for the registration and verification of vehicle pseudonyms. The real identity is never stored in plaintext on the public ledger. Ordinary vehicles can only verify the existence and validity of a pseudonym, while identity resolution is restricted to authorized entities via PRBC. The functions in the contract are explained as follows:

- *register_p_n:* This function takes five parameters id, p, p', τ, $\langle x \rangle_{sk}$. It is used to register vehicle pseudonyms, and verifies three conditions: (1) the validity of the signature (line 5); (2) the validity of the new pseudonym's usage period (line 6); and (3) that the new pseudonym has not been registered and the old pseudonym has been registered (line 7). After these verifications, it stores the new pseudonym (line 10), disables the old pseudonym (line 11), and returns the "pseudonym successfully registered".
- *register_p_i:* This function takes two parameters p_i and id, and can only be invoked by IMBC to register the initial pseudonym of the vehicle. If the provided initial pseudonym has not been registered, it registers and stores the initial pseudonym (line 18), sets the initial pseudonym's usage period to 0, and returns the "successfully registered".
- *check_p_n:* This function takes the parameter p, and can be invoked by all vehicles to query whether p has been registered (line 23). If registered, it returns "registered", otherwise it returns "unregistered".

Algorithm 2 Contract 2—Pseudonym registration

1: **persistent variables:**
2: **maintained by vehicles:**
3: id, p, p', sk, τ ▷ p old pseudonym, p' new pseudonym, τ the lifetime of new pseudonym.
4: **function** $register\_p_n(id, p, p', \tau, \langle x \rangle_{sk})$ ▷ signature object: $x \leftarrow \{id, p, p', \tau\}$.
5: **if** $Vrfy(\langle x \rangle_{sk}) = 0$; **return** "invalid signature"
6: **else if** $\tau < 0$; **return** "invalid lifetime"
7: **else if** $M_{pseu}(p') \neq null$ *or* $M_{pseu}(p) = null$;
8: **return** "invalid pseudonym"
9: **else**
10: $M_{\text{pseu}}(p') \Rightarrow \{hid, \tau\}$; $hid \leftarrow H(id \,\|\, r)$
11: $M_{pseu}(p) = null$;
12: **return** "pseudonym successfully registered"
13: **end function**
14:
15: **function** $register\_p_i(p_i, id)$ ▷ used by IMBC only.
16: **if** $M_{pseu}(p_i) \neq null$;
17: **return** "invalid initial pseudonym"
18: **else** $M_{pseu}(p_i) \Rightarrow \{id, 0\}$;
19: **return** "successfully registered"
20: **end function**
21:
22: **function** $check\_p_n(p)$ ▷ used by vehicles to verify a pseudonym.
23: **if** $M_{pseu}(p) = null$; **return** "unregistered"
24: **else return** "registered"
25: **end function**

The PseudonymRevocation contract is deployed on the PRBC and is responsible for revoking vehicle pseudonyms. The function in the contract is explained as follows:

- *revoke_p:* This function takes four parameters id, p_r, γ, $\langle x \rangle_{sk}$. It is used to revoke the vehicle pseudonym, and verifies two conditions. One is that the validity of the signature (line 7). The other is that p_r has not been revoked (line 8). If the revocation reason is "illegal," it requires that the transaction initiator is an authority (line 10), stores the revoked pseudonym, and disables the node of the malicious vehicle in GoT (line 13). If the revocation reason is "expired," it simply stores the revoked pseudonym (line 16), and finally returns the "successfully revoked".

Algorithm 3 Contract 3—Pseudonym revocation

1: **persistent variables:**
2: **maintained by vehicles:**
3: id, p_r, γ, sk ▷ p_r pseudonym to be revoked, γ reasons for revocation include "expired" and "illegal".
4: **maintained by PRBC:**
5: $\mathscr{A}_a$ ▷ addresses of authorities.
6: **function** $revoke\_p(id, p_r, \gamma, \langle x \rangle_{sk})$ ▷ signature object: $x \leftarrow \{id, p_r, \gamma\}$.
7: **if** $Vrfy(\langle x \rangle_{sk}) = 0$; **return** "invalid signature"
8: **else if** $M_{rev}(p_r) \neq null$; **return** "invalid pseudonym"
9: **else if** $\gamma =$ "illegal";
10: **if** $\omega \neq \mathscr{A}_a$; ▷ ω is the message sender's address.
11: **return** "not authorities"
12: **else** $M_{rev}(p_r) \Rightarrow \{id, \gamma\}$;
13: $modifyGoT(id) =$"successfully modified";
▷ remove id of the revoked vehicle from GoT.
14: **return** "successfully revoked"
15: **else if** $\gamma =$ "expired"
16: $M_{rev}(p_r) \Rightarrow \{id, \gamma\}$ **return** "successfully revoked"
17: **end function**

4.3. Normal-case operation

The normal-case operations of EBDA are divided into six parts: vehicle initialization, vehicle registration, generating the GoT, vehicle pseudonym registration, message validation, and vehicle pseudonym revocation. Vehicles must possess a valid identity before they are allowed to operate on the road for the first time. Initially, vehicles are initialized using cryptographic methods to generate unique key pairs and identity claims. The newly generated vehicle identity must be registered, and a GoT is created based on the vehicle's affiliation before it can be utilized.

Vehicle initialization: The initialization phase consists of key generation and identity claim generation. Unlike living beings, vehicles do not possess unique biological traits for differentiation. Instead, vehicles can use cryptographic techniques to generate a unique identity claim (id which is a series of attributes).

- Step1: The vehicle randomly chooses a integer as its private key sk, after which it uses the elliptic curve algorithm to generate the corresponding public key $pk = sk \cdot G$ (where G is the base point of the elliptic curve).
- Step2: The vehicle uses the SHA-256 hash algorithm to hash its unique serial number v and public key pk. The hash value $id = H(v, pk)$ is used as the vehicle's identity claim.

Vehicle registration: After initialization, the vehicle must register with the IMBC via an RSU. For identity registration, the vehicle sends a request containing its real information, including v, lp, id, id_o, $\mathscr{A}_o$, and the signature $\langle H(v, lp, \mathscr{A}_o, id)\rangle_{sk}$, to the RSU. The RSU then invokes the function *Register* from Contract 1 to complete the registration.

Generating the GoT: In VANETs, when a vehicle is travelling on the road, the surrounding entities change in real-time and the vehicle only interacts with nearby entities. Therefore, a vehicle cannot become trustworthy by signing a trust agreement or by accepting votes from surrounding entities.

However, the GoT represents relationships between devices and entities, as well as between entities [18]. Figure 2 illustrates a GoT where V denotes a vehicle, and A, B, C represent the vehicle owner, vehicle manufacturer, and device manufacturer, respectively. A subordinate relationship is formed between the vehicle and the owner through pairing, while trust relationships among the vehicle owner, vehicle manufacturer, and device manufacturer are established via signed trust agreements. This setup enables vehicle identity verification by assessing the vehicle owner's reputation and the trust relationships with other entities [19,20].

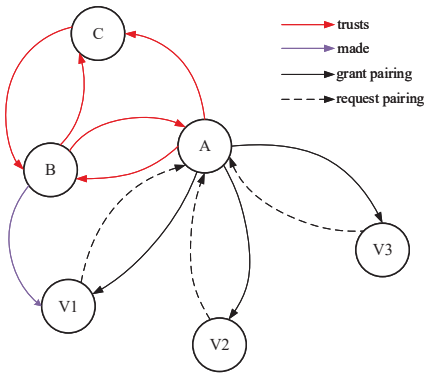


Figure 2. Sample GoT.

Generating a GoT involves the following steps:

- Step1: Based on the trust agreements between entities, the IMBC invokes the function *constructGoT* from Contract 1 to construct the GoT. For example, based on the trust relationship between the entities in Figure 2, IMBC uses $\{id_A, id_C\}$, $\{id_A, id_B\}$, $\{id_B, id_A\}$, $\{id_B, id_C\}$, $\{id_C, id_B\}$ as inputs to invoke the *constructGoT* to construct the GoT, respectively.
- Step2: IMBC uses the function *GoTaddnode* from Contract 1 to pair the vehicle's *id* with the vehicle owner's id_o, adding the vehicle to the GoT.

Pseudonym registratio: Pseudonyms address privacy and immediacy needs in VANET communication. We have developed a vehicle pseudonym generation algorithm and use a smart contract to handle pseudonym registration and verification services.

In the pseudonym generation algorithm, the vehicle takes its v, the vehicle timestamp μ and the vehicle's current latitude α and longitude β as inputs, and the vehicle's pseudonym as output. The algorithm proceeds as follows:

- Step1: The vehicle uses v and μ as inputs, a hash value is computed through SHA256 as a random number $q = H(v, \mu)$.
- Step2: In this step, a Pseudo-Random Number Generator (PRNG) [21] based on cross-coupled diagonal tent mappings is employed to generate a 256-bit pseudorandom sequence, as described below:

$$\begin{cases} x_{i+1} = f(y_i) \\ y_{i+1} = f(x_i) \end{cases} \qquad g = \begin{cases} 0 & f(x_i) \le f(y_i) \\ 1 & f(x_i) > f(y_i) \end{cases} \tag{1}$$

where f is the diagonal tent mapping defined in Equation (2), and g denotes the current output bit.

$$f(x) = x_{n+1} = \begin{cases} \dfrac{x_n}{p}, & x \in (0, p] \\ \dfrac{1 - x_n}{1 - p}, & x \in (p, 1) \end{cases} \tag{2}$$

where p is a control parameter close to 0.5. The PRNG is primarily seeded with a cryptographically secure value derived from the vehicle's secret key and timestamp, *i.e.*,

$$seed = H(sk \parallel \mu),$$

which ensures unpredictability against adversaries.

To introduce contextual diversity, the vehicle's latitude α and longitude β are normalized as follows:

$$\begin{cases} \alpha' = \dfrac{\alpha - (-90 - \varepsilon)}{180 + 2\varepsilon} \\ \beta' = \dfrac{\beta - (-180 - \varepsilon)}{360 + 2\varepsilon} \end{cases} \tag{3}$$

where ε is a small offset. The normalized coordinates (α', β') are incorporated only as auxiliary non-secret inputs to diversify the PRNG output, while the security of the generated 256-bit random sequence k does not rely on the secrecy or unpredictability of location information.

- Step3: The point M on the elliptic curve is obtained by multiplying q with the curve's generator G, expressed as $M = q \cdot G$.

- Step4: The random number k obtained in Step2 is XORed with the horizontal coordinate x_m of M obtained in Step3. The hash value of the XOR result is then used as the vehicle's new pseudonym $p = H(x_m \oplus k)$.

When a vehicle wants to register a new pseudonym p' on the PBC, it must send a request to the nearest RSU. This request includes the vehicle's id, the old pseudonym p, the new pseudonym p', the validity period τ of p', and the signature $\langle H(p, p', \tau, id)\rangle_{sk}$. Upon receiving the request, the RSU invokes the function *isTrustworthy* of Contract 1 to establish a trust relationship with the vehicle. The RSU then packages the request into a transaction and invokes the function $register\_p_n$ of Contract 2 to complete the pseudonym registration.

Message validation: To protect vehicle privacy, vehicles use pseudonyms for V2V and V2I communications. Each vehicle digitally signs the secure message with its private key and appends its valid pseudonym before transmitting the message, as shown in Equation (4).

$$V_{\mathrm{S}} \rightarrow \mathrm{X} : \mathrm{M}, \langle S_{\mathrm{M}} \mid T_{\mathrm{S}}\rangle_{sk}, p \tag{4}$$

where V_{S} is denoted as the sending-vehicle, X indicates the group of message receivers, M is the sending-message, $\mid$ represents the concatenation operation, T_{S} is the timestamp which ensures a received message as fresh message and finally, p is the vehicle's valid pseudonym. In V2V communication, when a vehicle receives a message from another vehicle, it invokes the $check\_p_n$ function to verify whether p is registered. If p is registered, the sender's message is deemed valid.

Pseudonym revocation: Vehicle pseudonyms are revoked either when the pseudonym's usage period expires or if the vehicle engages in malicious behavior. In the first case, the pseudonym owner sends a revocation request containing the pseudonym p_r, the reason γ, the vehicle's id, and the signature $\langle H(p_r, \gamma, id)\rangle_{sk}$ to the nearest RSU. The RSU then invokes the function $revoke\_p$ on Contract 3 to revoke the pseudonym. In the second case, violations are reported to an authority (e.g., traffic police), who initiates the revocation request. To mitigate revocation latency during network congestion, RSUs maintain a local revocation cache, allowing immediate suppression of malicious pseudonyms prior to on-chain confirmation.

PRBC records all pseudonyms used by each vehicle, allowing only authorities to trace a malicious vehicle's initial pseudonym back to PRBC to reveal its real information.

The real information query: When a vehicle commits malicious behavior, the authority first revokes its pseudonym and then retrieves the initial pseudonym p_i from PRBC. It then invokes the function *queryInfo* of Contract 1 using p_i as input.

We emphasize that the initialization of trust relationships in GoT does not introduce a centralized root of trust. The trust agreements are established among independent entities (e.g., manufacturers, owners, and device providers) through mutually signed statements, rather than being issued by a single authority. These agreements are recorded on-chain via smart contracts, making the trust graph transparent, verifiable, and auditable by all participants. No entity possesses unilateral power to define or override global trust relationships.

Moreover, GoT initialization is a one-time or low-frequency governance process, decoupled from time-critical authentication. During normal operation, vehicle authentication relies solely on existing trust paths in the GoT and does not require any centralized decision-making or online authority involvement.

5. Security analysis

In this section, we analyze the security of our EBDA. In particular, we show that EBDA satisfies all the security requirements listed in Section 3.3.

- Single registration: EBDA stores vehicle information on the blockchain during registration and adds the vehicle's identity claims to the GoT. RSUs then authenticate vehicles by querying the GoT, fulfilling the single registration requirement.
- Message authentication: RSUs or vehicles authenticate messages by verifying signatures and using smart contracts to check message integrity and pseudonym validity. Due to the elliptic curve discrete logarithm problem and blockchain's immutability, adversaries cannot forge valid messages in polynomial time.
- Conditional privacy preservation: The vehicle pseudon-ym generation algorithm relies on the random numbers from the PRNG. To test the randomness of the generated random number, we use the Statistical Test Suite by the National Institute of Standards and Technology (NIST) on a 10^7 bit sequence generated by Equation (1). As shown in Table 3, the sequence passed all tests, ensuring pseudonym non-forgeability. Vehicles periodically change pseudonyms, preventing adversaries from linking them to the vehicles' real identities. Access control is managed via smart contracts, with the TA revealing a vehicle's identity only in cases of malicious behavior.

Table 3. NIST test results.

Test	Result
Frequency	passed
BlockFrequency	passed
CumulativeSums	passed
Runs	passed
LongestRun	passed
Rank	passed
FastFourierTransform	passed
NonOverlappingTemplate	passed
OverlappingTemplate	passed
Universal	passed
ApproximateEntropy	passed
RandomExcursions	passed
RandomExcursionsVariant	passed
Serial	passed
LinearComplexity	passed

- Resistance to replay attack: To ensure the freshness of messages, vehicles include a timestamp and periodically change their pseudonyms. RSUs and vehicles can detect replay attacks by verifying message freshness and pseudonym validity, enabling EBDA to resist replay attacks.
- Resistance to man-in-the-middle attack: Since blockchain data is immutable and vehicle pseudonyms are non-forgeable, adversaries cannot forge pseudonyms in polynomial time. Additionally, valid messages are signed with the vehicle's private key, preventing impersonation.

Thus, EBDA is secure against the man-in-the-middle attacks.

- Resistance to DDos attack: The distributed blockchain structure in EBDA reduces vulnerability to DDoS attacks by making it hard to target all nodes. Besides, smart contracts handle pseudonym queries, ensuring stable operations even if some nodes are attacked. Therefore, EBDA withstands DDoS attacks.

6. Evaluation

In this section, we implement EBDA, emulating vehicle and pseudonym registration, verification, and revocation, and compare their performance with TBAA [7] and BPAS [14]. BPAS uses a certificate-free method similar to our approach, while TBAA combines blockchain with traditional PKI, recording transaction data on the blockchain to achieve traceable VANET access authentication. All reported verification latencies correspond to off-chain cryptographic verification and local contract state queries, rather than on-chain transaction confirmation delays.

In addition to BPAS and TBAA, Table 1 summarizes several certificate-free and blockchain-based authentication schemes, including [10] and [11]. These studies primarily focus on architectural design and security properties, rather than reporting reproducible end-to-end authentication latency under unified experimental settings. Moreover, such schemes typically involve multiple cryptographic operations or additional on-chain coordination during authentication. In contrast, EBDA restricts blockchain interactions to low-frequency control-plane procedures and executes time-critical authentication entirely off-chain through GoT queries and signature verification. Therefore, the quantitative evaluation in this work focuses on BPAS and TBAA, which provide explicit performance metrics and are widely adopted as representative baselines in blockchain-assisted VANET authentication.

6.1. Performance evaluation: setup and methodology

The EBDA is implemented through a prototype involving both blockchain and VANET simulation. This section delineates the specifics of the environment setup, and the ensuing performance evaluation affirms the effectiveness of the scheme.

We simulate the Ethereum network and VANET with Ganache, Veins, OMNet++, and SUMO on a laptop with an 8 vCore AMD Ryzen 5800H processor and 16 GB of RAM, respectively. Vehicle IDs and key pairs are generated with OpenSSL 3.1.4. Smart contracts are written in Solidity 0.8.0 and are deployed on the simulated Ethereum network. In the VANET simulations, we establish general urban roadway scenarios with vehicle speeds ranging from 10 to 20 miles per second, while the density representing the number of vehicles varies from 10 to 50 in increments of 10.

To evaluate our scheme's effectiveness, we conduct experiments on a range of performance metrics. We measure latency for vehicle registration and identity verification with a single vehicle owner. In addition, we analyze the latency for vehicle identity verification in a GoT with different numbers of vehicle owners. We also assess the latency for pseudonym registration, verification, and revocation under different scenarios on PBC and PRBC. The average values are calculated to ensure the reliability of the results. The reported verification latency measures the local execution time of cryptographic verification and contract state queries on a simulated environment, excluding blockchain propagation and consensus

delays. To provide an objective evaluation, we compare our scheme's performance with existing solutions, namely TBAA and BPAS, focusing on vehicle registration and verification latencies.

It is important to clarify that EBDA does not rely on real-time on-chain transactions for high-frequency VANET authentication. Blockchain interactions are limited to low-frequency control-plane operations such as vehicle registration, pseudonym issuance, and revocation. Time-critical authentication is performed entirely off-chain through local pseudonym verification and GoT-based trust checking, which incurs millisecond-level latency and is independent of blockchain confirmation delays or gas costs. As a result, the performance of EBDA is not constrained by Ethereum mainnet latency, and its scalability in dense urban scenarios primarily depends on off-chain verification efficiency rather than blockchain throughput.

6.2. Performance evaluation: results

Performance of registration with varying density of vehicles. Figure 3 shows the latency of vehicle registration for our scheme, TBAA, and BPAS. Our scheme and BPAS store vehicle identities on the blockchain, requiring only one registration over a vehicle's lifetime. In contrast, TBAA requires multiple registrations due to limited certificate validity. Consequently, TBAA's registration time is adjusted to reflect the estimated number of certificate renewals. Our scheme has an average registration latency of 0.175 s, which varies slightly with the number of vehicles (e.g., 0.171 s for $n = 30$ and 0.174 s for $n = 50$). TBAA and BPAS have average latencies of 0.181 s and 2.466 s, respectively, and BPAS has the highest latency due to the need for multiple parties to complete the registration process. Our scheme shows lower registration latencies compared to the other two, indicating greater efficiency in vehicle registration.

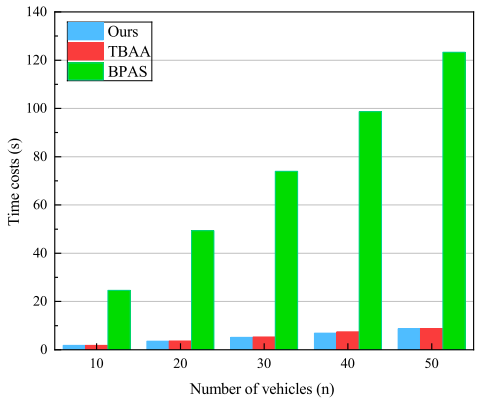
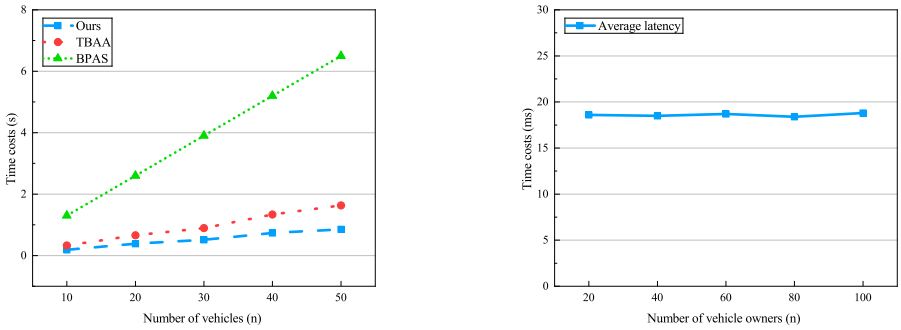


Figure 3. Comparison results for vehicles registration.

Performance of verification with varying density of vehicles. The latency for RSU to verify vehicle identities is shown in Figure 4a. In terms of the average verification latency per vehicle, our scheme demonstrates a significantly shorter latency of 18.06 ms, compared to 32.36 ms in TBAA and 130 ms in BPAS. The results indicate that our scheme significantly reduces the verification time, due to the use of the GoT for identity verification, enabling the RSU to quickly establish trust relationships by querying the GoT on the blockchain. In our scheme, vehicles only need to communicate with the RSU once during the entire authentication process. In contrast, TBAA requires three communications between the vehicle and RSU to complete the authentication, and the RSU also needs to communicate with the blockchain and

cloud servers, further increasing vehicle verification latency. The higher verification latency in BPAS is due to the extensive time required for complex cryptographic operations during the authentication process.

Performance of verification with varying number of owners. The latency for RSUs to verify vehicle identities with varying number of vehicle owners is depicted in Figure 4b. In this scenario, each vehicle owner is associated with a single vehicle, and the number of vehicle owners in the GoT is incremented. From the results shown in Figure 4b, it can be observed that as the number of vehicle owners in the GoT increases, the latency for verifying a single vehicle does not exhibit significant changes. This indicates that the performance of our scheme in verifying vehicle identities remains consistent and does not degrade with an increasing number of users (e.g., 18.6 ms for $n = 20$ and 18.8 ms for $n = 100$).



(a) **(b)**

Figure 4. Local verification execution latency. **(a)** Comparison results for vehicles verification; **(b)** Evaluation results for vehicle verification.

Evaluation of pseudonyms with varying density of vehicles. In EBDA, the average latencies for vehicle pseudonym registration, verification, normal revocation, and malicious vehicle pseudonym revocation are 290 ms, 17.2 ms, 149.47 ms, and 163.56 ms, respectively, as shown in Figure 5. The results indicate that our scheme can efficiently manage pseudonyms, providing fast registration, verification, and revocation processes across different scales. Specifically, as the number of vehicles increases, the time required for pseudonym registration, verification and revocation grows linearly. Additionally, when comparing abnormal situations to regular ones, the increase in time required for revocation is minimal, ensuring that our scheme can quickly respond to special situations without significant performance degradation.

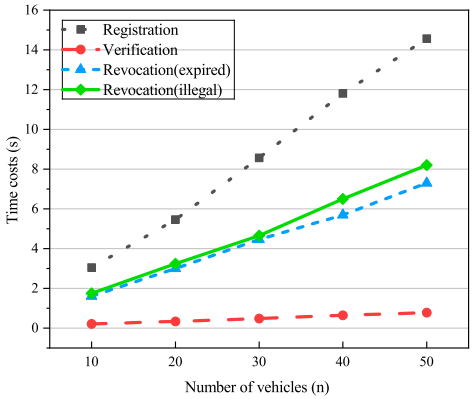


Figure 5. Performance of registration, verification and revocation of pseudonyms.

7. Concluding remarks

In this paper, we have proposed an Ethereum-based fully distributed vehicular authentication mechanism. Our EBDA addresses the centralization and inefficiency issues of PKI-based methods. It leverages blockchain and smart contracts to ensure secure and efficient authentication. Three distinct smart contracts are deployed on separate blockchains to manage vehicle identities and pseudonyms. Using GoT for identity verification eliminates certificates, reducing overhead and improving performance. We have conducted extensive experiments using a simulated Ethereum network and Veins. The results show that our approach outperforms existing schemes in terms of vehicle registration and verification latency. This proves its feasibility for real-world deployment. In future work, the GoT will be extended to incorporate reputation-based adaptive trust management. This will allow dynamic revocation of malicious vehicle identities without relying on TAs during the revocation phase. Additionally, we will add cross-domain authentication to enable secure interactions between vehicles and the external Internet of Things.

Data availability statement

No supplementary or additional data were generated in this study.

Declaration of generative AI and AI-assisted technologies

The authors did not use generative AI or AI-assisted technologies in the writing of this manuscript.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China General Program under Grant 62572220, in part by the Young Scientists Fund of the National Natural Science Foundation of China under Grant 62302202, in part by the Scientific Research Funds of Double World-Class Project 561120208, in part by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China under Grant 62121001, in part by the Key Program of NSFC under Grant U1405255.

Authors' contribution

Conceptualization, Chunyan Liu and Xiaoqin Feng; methodology, Fuliang Lin and Tao Feng; software, Fuliang Lin; validation, Chunyan Liu and Tao Feng; formal analysis, Hongkun Tian; investigation, Fuliang Lin; resources, Xiaoqin Feng; data curation, Chunyan Liu; writing—original draft preparation, Chunyan Liu and Fuliang Lin; writing—review and editing, Hongkun Tian and Tao Feng; visualization, Fuliang Lin; supervision, Xiaoqin Feng; project administration, Chunyan Liu; funding acquisition, Xiaoqin Feng. All authors have read and agreed to the published version of the manuscript.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Moussaoui D, Kadri B, Feham M, Bensaber BA. A distributed blockchain based PKI (BCPKI) architecture to enhance privacy in VANET. In *Proceedings of 2nd IEEE International Workshop on Human-Centric Mmart Environments for Health and Well-being (IHSH)*, Dubai, United Arab Emirates, June 21–23, 2021, pp. 75–79.

[2] Jiang S, Chen X, Cao Y, Xu T, He J, *et al.* APKI: an anonymous authentication scheme based on PKI for VANET. In *Proceedings of 7th International Conference on Computer and Communication Systems (ICCCS)*, Wuhan, China, April 22–25, 2022, pp. 530–536.

[3] Wang F, Xu Y, Zhang H, Zhang Y, Zhu L. 2FLIP: a two-factor lightweight privacy-preserving authentication scheme for VANET. *IEEE Trans. Veh. Technol.* 2016, 65(2):896–911.

[4] Li X, Liu T, Obaidat MS, Wu F, Vijayakumar P, *et al.* A lightweight privacy-preserving authentication protocol for VANETs. *IEEE Syst. J.* 2020, 14(3):3547–3557.

[5] Calandriello G, Papadimitratos P, Hubaux JP, Lioy A. Efficient and robust pseudonymous authentication in VANET. In *Proceedings of the 4th ACM International Workshop on Vehicular Ad Hoc Networks*, Montréal, Canada, September 10, 2007, pp. 19–28.

[6] Zhong H, Han S, Cui J, Zhang J, Xu Y. Privacy-preserving authentication scheme with full aggregation in VANET. *Inf. Sci.* 2019, 476:211–221.

[7] Zheng D, Jing C, Guo R, Gao S, Wang L. A traceable blockchain-based access authentication system with privacy preservation in VANETs. *IEEE Access* 2019, 7:117716–117726.

[8] Dwivedi SK, Amin R, Vollala S, Das AK. Design of blockchain and ECC-based robust and efficient batch authentication protocol for vehicular ad-hoc networks. *IEEE Trans. Intell. Transp. Syst.* 2024, 25(1):275–288.

[9] Lu Z, Wang Q, Qu G, Zhang H, Liu Z. A blockchain-based privacy-preserving authentication scheme for VANETs. *IEEE Trans. Very Large Scale Integr. VLSI Syst.* 2019, 27(12):2792–2801.

[10] Akhter AS, Ahmed M, Shah AS, Anwar A, Kayes A, *et al.* A blockchain-based authentication protocol for cooperative vehicular ad hoc network. *Sensors* 2021, 21(4):1273.

[11] Li H, Pei L, Liao D, Sun G, Xu D. Blockchain meets VANET: an architecture for identity and location privacy protection in VANET. *Peer-to-Peer Networking Appl.* 2019, 12:1178–1193.

[12] Feng Q, He D, Zeadally S, Liang K. BPAS: blockchain-assisted privacy-preserving authentication system for vehicular ad hoc networks. *IEEE Trans. Ind. Inf.* 2019, 16(6):4146–4155.

[13] Chattaraj D, Bera B, Das AK, Saha S, Lorenz P, *et al.* Block-CLAP: blockchain-assisted certificateless key agreement protocol for internet of vehicles in smart transportation. *IEEE Trans. Veh. Technol.* 2021, 70(8):8092–8107.

[14] Feng Q, He D, Zeadally S, Liang K. BPAS: blockchain-assisted privacy-preserving authentication system for vehicular ad hoc networks. *IEEE Trans. Ind. Inf.* 2020, 16(6):4146–4155.

[15] Javaid U, Aman MN, Sikdar B. DrivMan: driving trust management and data sharing in VANETs with blockchain and smart contracts. In *Proceedings of 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, Kuala Lumpur, Malaysia, April 28–May 1, 2019, pp. 1–5.

[16] Wei L, Cui J, Zhong H, Bolodurina I, Gu C, *et al.* A decentralized authenticated key agreement

scheme based on smart contract for securing vehicular ad-hoc networks. *IEEE Trans. Mob. Comput.* 2023, 23(5):4318–4333.

[17] Kchaou A, Ayed S, Abassi R, El Fatmi SG. Smart contract-based access control for the vehicular networks. In *Proceedings of International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Split, Croatia, September 17–19, 2020, pp. 1–6.

[18] Schaerer J, Zumbrunn S, Braun T. Veritaa-IoT: a distributed public key infrastructure for the Internet of Things. In *Proceedings of IFIP Networking Conference (IFIP Networking)*, Catania, Italy, June 13–16, 2022, pp. 1–9.

[19] Liu Z, Wan L, Guo J, Huang F, Feng X, *et al.* PPRU: a privacy-preserving reputation updating scheme for cloud-assisted vehicular networks. *IEEE Trans. Veh. Technol.* 2023, 74(2):1877–1892.

[20] Liu Z, Weng J, Ma J, Guo J, Feng B, *et al.* TCEMD: a trust cascading-based emergency message dissemination model in VANETs. *IEEE Internet Things J.* 2020, 7(5):4028–4048.

[21] Elmanfaloty RA, Abou-Bakr E. Random property enhancement of a 1D chaotic PRNG with finite precision implementation. *Chaos, Solitons Fractals* 2019, 118:134–144.