EPiC Series in Computing

Volume 106, 2025, Pages 84–98

Proceedings of the 20th Asia Joint Conference on Information Security

Traceable Cross-Domain Anonymous Authentication for Distributed Vehicular Fog Computing Services

Hai Liang[1], Wenkang Tao[1], Yujue Wang[3,*], Shuo Wang[1], Yu Xu[1], and Xinyong Peng[2]

[1] Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin, China

[2] Guangxi Key Laboratory of Digital Infrastructure, Guangxi Zhuang Autonomous Region Information Center, Nanning, China

[3] Hangzhou Innovation Institute of Beihang University, Hangzhou, China

Abstract

With the development of distributed vehicle fog services (VFS), the demand for vehicle authentication in high-speed mobility and cross-domain scenarios has grown significantly. However, traditional authentication schemes exhibit significant limitations in privacy protection, low latency, and multi-domain collaborative authentication, particularly in high-speed scenarios where vehicles frequently switch domains. Additionally, existing solutions that rely on centralized authentication architectures are vulnerable to single points of failure, further exacerbating security risks.To address these challenges, this paper proposes a Blockchain-Assisted Traceable Cross-Domain Anonymous Authentication Mechanism (BTCAA), aimed at providing secure and efficient authentication for high-speed moving vehicles accessing VFS. BTCAA designs a flexible authentication process that allows vehicles to dynamically adjust authentication procedures based on their driving routes and introduces anonymity to protect user privacy. The mechanism adopts a lightweight design to reduce authentication overhead while supporting identity traceability, ensuring the ability to verify vehicle identities in dispute scenarios without compromising anonymity. The decentralized architecture eliminates the risk of single point of failure, improving the security of the system. The security analysis confirms that BTCAA effectively ensures privacy protection, identity traceability, message integrity, and confidentiality. Performance evaluations further demonstrate its high practicality and efficiency while maintaining robust security and privacy protection.

Keywords: Anonymous, Blockchain, Cross Domain Authentication, Vehicle Fog Services

1 Introduction

VANETs, a subset of MANETs, consist of vehicles, Roadside Units (RSUs), and On-Board Units (OBUs), and rely on cloud computing for communication, computation, and storage [1]. As connected vehicles increase, the demand for uninterrupted low-latency services increases, which presents challenges in integrating traditional cloud computing with VANET-based vehicular cloud services [2].

Vehicular Fog Computing (VFC) aims to address these challenges by providing distributed computing resources near vehicles to reduce latency and improve service efficiency [3]. However, VFC faces security and performance issues such as vehicle identity authentication, privacy protection, and dispute resolution [4].

Cross-domain identity authentication is crucial, but exposes significant security and privacy risks, such as unauthorized access and impersonation attacks. Secure, efficient, and lightweight authentication mechanisms are necessary to protect privacy and enable traceable identities when needed [5].

Existing authentication mechanisms, including symmetric encryption and public key infrastructures, struggle in cross-domain scenarios due to reliance on vulnerable centralized databases [6]. The decentralized nature, immutability, and consensus mechanisms of the blockchain offer a promising solution, reducing communication overhead, eliminating database dependency, and enabling traceable identity management in latency-sensitive vehicular services [7].

1.1 Contributions

To address the significant challenges in vehicular networks, including limitations in privacy protection, high latency, inefficient multi-domain authentication, and the security risks of single points of failure inherent in centralized architectures, this paper proposes a Blockchain-Assisted Traceable Cross-Domain Anonymous Authentication Mechanism (BTCAA). By leveraging blockchain's decentralized nature, the mechanism reduces cross-domain authentication latency and enhances security through immutability and distributed storage. It ensures vehicle identity anonymity and prevents attackers from tracking behavior. Additionally, it supports efficient identity tracing during disputes, balancing security and performance.

Our BTCAA construction offers the following features:

1. **Flexible Cross-Domain Authentication**: Vehicles dynamically generate Lagrange basis functions based on Service Manager (SM) information, allowing selected domains to extract necessary data for authentication securely.

2. **Anonymity for Privacy Protection**: Vehicles use pseudonyms generated by the Audit Department (AD) to enhance privacy protection, with one real vehicle ID corresponding to multiple pseudonyms.

3. **Efficient Authentication Process**: Vehicles send a single message during movement, either containing authentication or service request information. In case of a dispute, identity tracing can be initiated to recover the real identity through encryption.

4. **Secure, Transparent Authentication**: Public authentication data are stored on the blockchain, ensuring transparency, resistance to tampering, and security against common attacks.

1.2 Organization

The remainder of this paper is organized as follows. Section 2 introduces related works. Section 3 describes the system model and security requirements. Section 4 provides a detailed explanation of the system design. Security analysis and performance evaluation are presented in Sections 5 and 6, respectively. Finally, Section 7 summarizes the paper and discusses future work.

2 Related Works

Anonymous authentication is essential for privacy in VANETs [6]. Zhang et al. [8] proposed a blockchain-based vehicular authentication protocol offering anonymous authentication and key agreement with lightweight operations, improving resistance to known attacks. Duan et al. [9] integrated blockchain with unclonable functions to reduce computational overhead while maintaining anonymity and non-repudiation. Despite these advances, challenges remain, including reliance on trusted third parties and high requirements of node performance.

Wang et al. [10] developed a cloud-based road condition monitoring scheme without considering the vulnerabilities to centralized failures. Da et al. [11] improved this with a blockchain-assisted scheme featuring pseudonym mechanisms for decentralization and distributed blockchain supervision to detect server dishonesty, which optimizes efficiency using elliptic curve cryptography without sacrificing security. These advancements, alongside Fan's DAFL [12] and Mohammed's ANAA-Fog [13], move toward fully decentralized and lightweight authentication solutions.

Traditional centralized cross-domain authentication in VANETs is prone to attacks and scalability issues [14]. Bartsch et al. [15] proposed a PKI-based scheme but faced high overhead and lack of flexibility. Cui et al. [16] and Wang et al. [17] similarly rely on centralized facilities, leading to single-point failures.

Blockchain-based schemes address these issues by offering decentralization and transparency. Zhang et al. [8] combined blockchain with PKI but faced efficiency issues in large-scale settings. Liu et al. [18] designed a blockchain-based IoT data-sharing scheme, though key and pseudonym updates are complex. Xue et al. [19] optimized authentication with distributed certificates but at the cost of increased communication overhead. Feng et al.'s proposal [20] for 5G environments incurred high communication overhead from multiple interaction rounds, while Deng et al.'s scheme [21] for V2G networks was hampered by complex certificate management and large storage demands.

Chen et al.'s identity system [22] was limited by its high resource demands on vehicles, while Shen et al.'s framework [23] incurred significant communication overhead and latency from server-blockchain interactions. Therefore, while blockchain improves security and decentralization, further optimization is needed to address communication overhead, storage, and scalability in large-scale vehicular networks.

3 System Overview

3.1 System Model

As shown in Figure 1, a BTCAA system consists of five types of entities, namely AD, SM, OBU, Data Service Center, and Consortium Blockchain Network.

1) **AD**: AD is a fully trusted entity responsible for registering OBU and SM, as well as tracking unauthorized vehicles.

2) **SM**: SM is a regional service manager that must be registered with AD before the system is established. SM is primarily responsible for managing all Vehicular Fog datacenters (VFDs) and authenticated OBUs within its region.

3) **OBU**: OBU is the primary user of the system. It possesses computational and communication capabilities, such as an embedded computer, wireless network interface, GPS receiver, vehicle navigation system, and digital maps. It must be registered with AD before accessing the system.

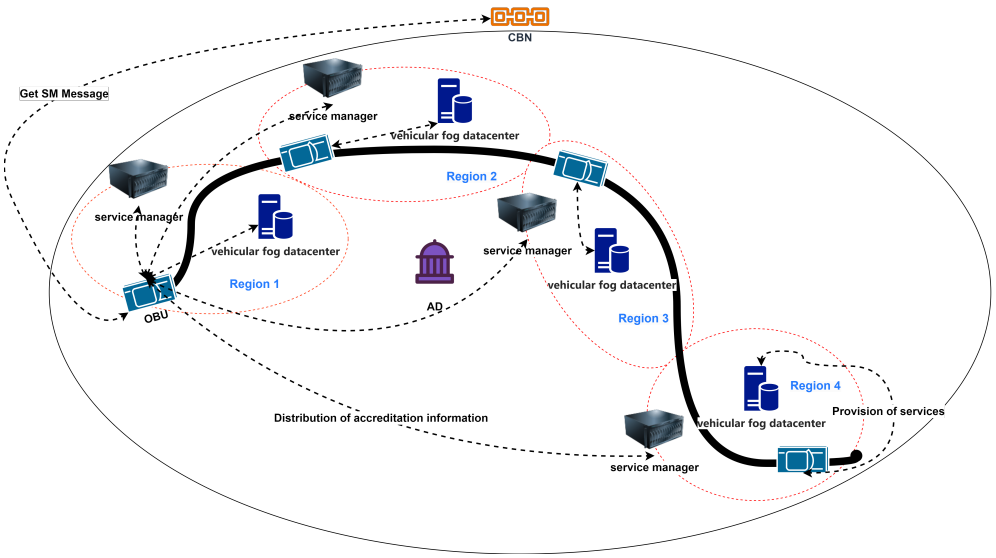


Figure 1: System Model

4) **VFD**: VFD is a regional data center. It is primarily responsible for providing the required data services to authenticated OBUs.

5) **Consortium Blockchain Network (CBN)**: The CBN consists of all regional SMs working together as a decentralized, tamper-proof distributed ledger. It is primarily used to securely store and distribute system-wide public parameters and public key information of registered SMs. Through the blockchain, consistent and trusted access to this critical data can be ensured to all entities, thus eliminating the dependence on centralized servers and enhancing the robustness and attack resistance of the system.

In the BTCAA system, cross-domain authentication is required when a user needs local data services from different regions. First, the user registers with the AD, which generates public/private keys and pseudonym sequences, binding the real identity to restoration parameters stored locally. The user uses their own keys and pseudonyms, along with the public keys and ID information from the SMs of the domains to be crossed, to generate authentication parameters, which are then distributed to the subsequent SMs. The SM performs authentication, and if successful, notifies the local data service center to provide services. If the data service center questions the user's identity or behavior, it sends the pseudonym to the AD. The AD compares it with the stored restoration parameters, and if a match is found, the real identity is confirmed. Inappropriate behavior may result in penalties. If no match is found, the SM is notified of the failure and may refuse further services.

3.2 Security Requirements

Before detailing the security requirements, we first define the threat model of the system. We assume that an attacker is able to listen to, intercept, modify, and replay all communication messages on the public channel. However, we assume that an attacker is unable to compromise a highly trusted AD or solve cryptographic problems such as the ECDLA in polynomial time.

Identity Anonymity: To protect the identity privacy during the authentication process, the real identity of the vehicle must be replaced with an anonymous identity. Other entities, except for the AD, cannot trace the vehicle's real ID.

Traceability: During cross-domain authentication or data service provision, if the SM questions the behavior or real identity of a cross-domain device, it can immediately send a trace-back request to the AD. The AD should be able to restore the real identity of the cross-domain device.

Confidentiality: Apart from the authentication initiator (OBU) and the recipient (SM), other vehicle users or attackers cannot obtain the real ID of the OBU or any valid parameters related to the authentication process.

Message Integrity: The system must ensure that encrypted messages during the registration phase and all parameters transmitted during the authentication phase are not tampered with. Even in the event of an attack, tampering behavior can be quickly detected and identified.

Resistance to Replay Attacks: During the authentication process, the system must be able to prevent attackers from replaying legitimate messages in an attempt to impersonate legitimate devices. This will ensure that each authentication request is unique. Even if an attacker replays a previous message, it should fail the validation, thereby effectively mitigating replay attacks.

4 BTCAA Construction

This section presents the detailed design of the BTCAA system, which consists of five stages: System Initialization, SM Registration, OBU Registration, Authentication, and Tracking. The security of our BTCAA construction relies on the following two computational assumptions.

Elliptic Curve Computational Diffie-Hellman Assumption (ECDHA): Let G be a cyclic group of points on an elliptic curve with prime order q, and P be a generator of G. Given random points $Q_1 = aP$, $Q_2 = bP$, and P, where $a, b \in \mathbb{Z}_q^*$, it is computationally difficult to output abP.

Elliptic Curve Discrete Logarithm Assumption (ECDLA): Let G be an additive group of points on an elliptic curve with prime order q, and P be a generator of G. Given $xP \in G$, it is computationally difficult to compute x.

4.1 System Initialization

The AD generates a large safe prime number q, and selects an elliptic curve cyclic group G of order q, along with its generator P. The following hash functions are defined: $H_0 : \{0,1\}^* \times G \times G \to \mathbb{Z}_q^*$, $H_1 : G \times \{0,1\}^* \times \{0,1\}^* \to \mathbb{Z}_q^*$, $H_2 : G \to \{0,1\}^{256}$, $H_3 : \{0,1\}^* \to \mathbb{Z}_q^*$. Next, the AD randomly selects an element $\mathrm{SK_{AD}} \in \mathbb{Z}_q^*$ as its private key and computes the public key $\mathrm{PK_{AD}} = \mathrm{SK_{AD}} \cdot P$. The AD keeps the private key $\mathrm{SK_{AD}}$ confidential and records the system public parameters $(q, \mathrm{PK_{AD}}, H_0, H_1, H_2, H_3, P)$ on the blockchain. In addition, Select a secure symmetric encryption, denoted as Enc_X, where X represents the symmetric key.

4.2 SM Registration Phase

This phase completes the registration of the SM as follows:

1. The SM randomly selects an element $\mathrm{SK}^1_{\mathrm{SM}_i} \in \mathbb{Z}_q^*$ as the first part of its private key and computes the first part of the public key $\mathrm{PK}^1_{\mathrm{SM}_i} = \mathrm{SK}^1_{\mathrm{SM}_i} \cdot P$. Then, it sends the first part of the public key along with its ID $(\mathrm{PK}^1_{\mathrm{SM}_i}, \mathrm{SM}_i)$ to the AD.

2. Upon receiving $(\mathrm{PK}^1_{\mathrm{SM}_i}, \mathrm{SM}_i)$, the AD checks the ID and computes $K_{sharedSM} = H_2(\mathrm{SK_{AD}} \cdot \mathrm{PK}^1_{\mathrm{SM}_i})$. It then randomly selects an element $d_i \in \mathbb{Z}_q^*$

and computes $\mathrm{PK}^2_{\mathrm{SM}_i} = d_i \cdot P$ and $\mathrm{SK}^2_{\mathrm{SM}_i} = d_i + \mathrm{SK}_{\mathrm{AD}} \cdot H_0(\mathrm{SM}_i, \mathrm{PK}^2_{\mathrm{SM}_i}, \mathrm{PK}^1_{\mathrm{SM}_i})$. The AD then encrypts $\mathrm{SK}^2_{\mathrm{SM}_i}||\mathrm{PK}^2_{\mathrm{SM}_i}$ using $K_{sharedSM}$ and returns the ciphertext $(\mathrm{Enc}_{K_{sharedSM}}(\mathrm{SK}^2_{\mathrm{SM}_i}||\mathrm{PK}^2_{\mathrm{SM}_i}))$.

3. Upon receiving the ciphertext $\mathrm{Enc}_{K_{sharedSM}}(\mathrm{SK}^2_{\mathrm{SM}_i}||\mathrm{PK}^2_{\mathrm{SM}_i})$, the SM computes $K_{sharedSM} = H_2(\mathrm{SK}^1_{\mathrm{SM}_i} \cdot \mathrm{PK}_{\mathrm{AD}})$ to obtain the symmetric key and decrypts to extract the second parts of the private key and public key. The SM then combines the first and second parts of the public and private keys, where the public key is represented as $\mathrm{PK}_{\mathrm{SM}_i} = (\mathrm{PK}^1_{\mathrm{SM}_i}, \mathrm{PK}^2_{\mathrm{SM}_i})$ and the private key as $\mathrm{SK}_{\mathrm{SM}_i} = (\mathrm{SK}^1_{\mathrm{SM}_i}, \mathrm{SK}^2_{\mathrm{SM}_i})$. The following verification is then performed to check the correctness of the key:

$$\mathrm{PK}^2_{\mathrm{SM}_i} + H_0(\mathrm{SM}_i, \mathrm{PK}^2_{\mathrm{SM}_i}, \mathrm{PK}^1_{\mathrm{SM}_i}) \cdot \mathrm{PK}_{\mathrm{AD}} \stackrel{?}{=} \mathrm{SK}^2_{\mathrm{SM}_i} \cdot P$$

If the verification is successful, the SM stores the complete public and private keys locally and records $(\mathrm{SM}_i, \mathrm{PK}^1_{\mathrm{SM}_i}, \mathrm{PK}^2_{\mathrm{SM}_i})$ on the blockchain.

4.3 OBU Registration Phase

This phase completes the registration of the OBU as follows:

1. The OBU randomly selects an element $\mathrm{SK}^1_{\mathrm{OBU}_i} \in \mathbb{Z}^*_q$ as the first part of its private key and computes $\mathrm{PK}^1_{\mathrm{OBU}_i} = \mathrm{SK}^1_{\mathrm{OBU}_i} \cdot P$, then computes $K_{sharedOBU} = H_2(\mathrm{SK}^1_{\mathrm{OBU}_i} \cdot \mathrm{PK}_{\mathrm{AD}})$ and sends the first part of the public key along with its symmetric encrypted ID

$$(\mathrm{PK}^1_{\mathrm{OBU}_i}, \mathrm{Enc}_{K_{sharedOBU}}(\mathrm{OBU}_i))$$

to the AD.

2. Upon receiving $(\mathrm{PK}^1_{\mathrm{OBU}_i}, \mathrm{Enc}_{K_{shared_{OBU}}}(\mathrm{OBU}_i))$, the AD computes the shared key $K_{sharedOBU} = H_2(\mathrm{SK}_{\mathrm{AD}} \cdot \mathrm{PK}^1_{\mathrm{OBU}_i})$, decrypts the ID, and generates n pseudonyms. First, it randomly selects an element $x \in \mathbb{Z}^*_q$ and computes $Y = x \cdot P$ and $PID_{i,j} = \mathrm{OBU}_i \oplus H_2(j \cdot Y),\ 1 \le j \le n$. Then, it randomly selects an element $d_i \in \mathbb{Z}^*_q$, computes $\mathrm{PK}^2_{\mathrm{OBU}_i} = d_i \cdot P$, and $\mathrm{SK}^2_{\mathrm{OBU}_{i,j}} = d_i + \mathrm{SK}_{\mathrm{AD}} \cdot H_0(\mathrm{PID}_{i,j}, \mathrm{PK}^2_{\mathrm{OBU}_i}, \mathrm{PK}^1_{\mathrm{OBU}_i}),\ 1 \le j \le n$. The AD then generates the ciphertext

$$C_{OBU} = \mathrm{Enc}_{K_{sharedOBU}}(\mathrm{SK}^2_{\mathrm{OBU}_{i,j}}||\mathrm{PK}^2_{\mathrm{OBU}_i}||\{(j, PID_{i,j}) : 1 \le j \le n\})$$

and returns it. The AD stores (OBU_i, Y) locally.

3. Upon receiving C_{OBU}, the OBU decrypts the ciphertext to extract the second parts of the public and private keys as well as the pseudonym information. The OBU then combines the two parts of the public and private keys to obtain the complete keys, with the complete public key represented as $\mathrm{PK}_{\mathrm{OBU}_i} = (\mathrm{PK}^1_{\mathrm{OBU}_i}, \mathrm{PK}^2_{\mathrm{OBU}_i})$ and the complete private key list as $\mathrm{SK}_{\mathrm{OBU}_{i,j}} = (\mathrm{SK}^1_{\mathrm{OBU}_i}, \mathrm{SK}^2_{\mathrm{OBU}_{i,j}}),\ 1 \le j \le n$, and the pseudonym list as $PID_{i,j},\ 1 \le j \le n$. The following verification is then performed to check the correctness of the keys:

$$\mathrm{PK}^2_{\mathrm{OBU}_i} + H_0(\mathrm{PID}_{i,j}, \mathrm{PK}^2_{\mathrm{OBU}_i}, \mathrm{PK}^1_{\mathrm{OBU}_i}) \cdot \mathrm{PK}_{\mathrm{AD}} \stackrel{?}{=} \mathrm{SK}^2_{\mathrm{OBU}_{i,j}} \cdot P, \quad 1 \le j \le n$$

If the verification is successful, the OBU stores the complete public and private keys along with the corresponding pseudonym list locally.

4.4 Authentication Phase

In this phase, the OBU generates authentication parameters based on the pre-determined route through various domains and distributes these parameters to all relevant domains. Within each domain, the SM verifies the identity of the OBU based on the calculated authentication parameters to decide whether to provide the corresponding services.

1. The OBU randomly selects a tuple

$$(\mathrm{PK}^1_{\mathrm{OBU}_i}, \mathrm{PK}^2_{\mathrm{OBU}_i}, \mathrm{SK}^1_{\mathrm{OBU}_i}, \mathrm{SK}^2_{\mathrm{OBU}_{i,j}}, j, \mathrm{PID}_{i,j}),$$

and randomly selects $r_{j,1} \in \mathbb{Z}_q^*$, then computes

$$c_{j,1} = r_{j,1} \cdot P \quad \text{and} \quad c_{j,2} = r_{j,1} + H_1(c_{j,1}, t_j, \mathrm{PID}_{i,j}) \cdot (\mathrm{SK}^1_{\mathrm{OBU}_i} + \mathrm{SK}^2_{\mathrm{OBU}_{i,j}}),$$

where t_j is the timestamp. Next, the OBU retrieves the public keys and IDs of the SMs in all target domains from the blockchain, and computes

$$h_i = H_0(\mathrm{SM}_i, \mathrm{PK}^2_{\mathrm{SM}_i}, \mathrm{PK}^1_{\mathrm{SM}_i}), 1 \le i \le k$$

where k is the total number of target domains. Then, the OBU randomly selects an element $r_1 \in \mathbb{Z}_q^*$ and computes $R = r_1 \cdot P$. To ensure consistent parameter group lengths for subsequent operations, $H_2(R)$ is used to fill a byte string L of the same length as the parameter group.

$$\mathrm{PID}_{i,j} \| c_{j,1} \| c_{j,2} \| t_j \| j \| \mathrm{PK}^1_{\mathrm{OBU}_i} \| \mathrm{PK}^2_{\mathrm{OBU}_i}.$$

Then, the byte-wise XOR operation is performed between L and the parameter group to compute

$$C = L \oplus (\mathrm{PID}_{i,j} \| c_{j,1} \| c_{j,2} \| t_j \| j \| \mathrm{PK}^1_{\mathrm{OBU}_i} \| \mathrm{PK}^2_{\mathrm{OBU}_i}).$$

Next, the OBU computes $x_i = H_3(\mathrm{SM}_i), 1 \le i \le k$. Then, the OBU constructs the following interpolation polynomial:

$$f_i(x) = \prod_{1 \le j \ne i \le k} \frac{x - x_j}{x_i - x_j} = u_{i,1} + u_{i,2}x + \cdots + u_{i,k}x^{k-1}, 1 \le i \le k.$$

Next, the OBU computes $Q_i = r_1 \cdot (\mathrm{PK}^1_{\mathrm{SM}_i} + \mathrm{PK}^2_{\mathrm{SM}_i} + h_i \cdot \mathrm{PK}_{\mathrm{AD}})$, and uses the coefficients $u_{i,1}, u_{i,2}, \ldots, u_{i,k} \in \mathbb{Z}_q^*$ to compute

$$V_i = \sum_{l=1}^{k} u_{l,i} Q_l, \quad 1 \le i \le k.$$

Finally, the OBU generates a set of authentication parameters $\delta = (V_1, V_2, \ldots, V_k, C)$, and send δ to the SMs in all participating target domains.

2. SM_v computes $x_v = H_3(\mathrm{SM}_v)$ and

$$Q'_v = V_1 + x_v V_2 + \cdots + x_v^{k-1} V_k.$$

Then, the SM_v computes $R' = (\mathrm{SK}^1_{\mathrm{SM}_v} + \mathrm{SK}^2_{\mathrm{SM}_v})^{-1} Q'_v$, and uses $H_2(R')$ to fill a byte string L' of the same length as C, which is then XORed with C to recover the original message:

$$(\mathrm{PID}_{i,j} \| c_{j,1} \| c_{j,2} \| t_j \| j \| \mathrm{PK}^1_{\mathrm{OBU}_i} \| \mathrm{PK}^2_{\mathrm{OBU}_i}) = L' \oplus C.$$

After the authentication parameters are restored, SM_v can use the obtained parameters to verify whether the identity of the OBU requesting cross-domain authentication is valid. The verification process is as follows:

$$c_{j,2} \cdot P \stackrel{?}{=} c_{j,1} + H_1(c_{j,1}, t_j, PID_{i,j}) \cdot \left(PK^1_{OBU_i} + PK^2_{OBU_i} + H_0(PID_{i,j}, PK^2_{OBU_i}, PK^1_{OBU_i}) \cdot PK_{AD}\right)$$

If the verification is successful, the SM notifies the VFD in its domain to provide services to the OBU.

In fact, when SM receives requests from multiple OBUs, BTCAA can also perform batch verification to improve verification efficiency. For each vehicle request tuple

$$\{(\mathrm{PID}_{i,j} \| c_{i,j,1} \| c_{i,j,2} \| t_j \| j \| \mathrm{PK}^1_{\mathrm{OBU}_i} \| \mathrm{PK}^2_{\mathrm{OBU}_i}) : 1 \le i \le l\},$$

a random element $\gamma_i \in \mathbb{Z}_p^*$ is selected. SM then uses each request tuple along with its corresponding random element to verify the legitimacy of each OBU's identity. The verification method is as follows:

$$\sum_{i=1}^{l} c_{i,j,2} \cdot \gamma_i \cdot P \stackrel{?}{=} \sum_{i=1}^{l} \gamma_i c_{i,j,1} + \sum_{i=1}^{l} H_1(c_{i,j,1}, t_j, \mathrm{PID}_{i,j}) \cdot \gamma_i \cdot (\mathrm{PK}^1_{\mathrm{OBU}_i} + \mathrm{PK}^2_{\mathrm{OBU}_i} + H_0(\mathrm{PID}_{i,j}, \mathrm{PK}^2_{\mathrm{OBU}_i}, \mathrm{PK}^1_{\mathrm{OBU}_i}) \cdot \mathrm{PK}_{\mathrm{AD}})$$

If the equation holds, the SM notifies the VFD in its domain to provide services for all OBUs in this batch.

4.5 Tracking phase

If the SM has doubts about the identity of the OBU during the authentication phase or about the behavior of the OBU during subsequent service phases, the SM can extract the tuple $(\mathrm{PID}_{i,j} \| c_{j,1} \| c_{j,2} \| t_j \| j \| \mathrm{PK}^1_{\mathrm{OBU}_i} \| \mathrm{PK}^2_{\mathrm{OBU}_i})$ and send $(j, \mathrm{PID}_{i,j})$ to the AD. The AD checks all (OBU_i, Y) and computes $\mathrm{OBU}'_i = \mathrm{PID}_{i,j} \oplus H_2(j \cdot Y)$. If $\mathrm{OBU}'_i = \mathrm{OBU}_i$, then the trace is successful. If any inappropriate behavior is found, the AD broadcasts a message to other SMs to suspend all cross-domain requests for this vehicle. If the trace fails, the AD returns the trace result to the SM. Upon receiving the trace result, the SM suspends the subsequent data services for this vehicle.

5 Security Analysis

5.1 Correctness Analysis

The correctness of the proposed mechanism relies on whether the receiver can correctly compute Y_v, thereby deriving the correct R, and then use the XOR operation with C to obtain the parameters required for subsequent authentication. Therefore, this section verifies the key exchange process during the registration phase, and also verifies whether $Y'_v = Y_v$ and $R' = R$ hold.

For the key exchange process during the registration phase:

$$K_{\mathrm{shared}} = H_2(\mathrm{PK}_{\mathrm{AD}} \cdot \mathrm{SK}^1_{\mathrm{SM}_i}) = H_2(\mathrm{SK}_{\mathrm{AD}} \cdot \mathrm{PK}^1_{\mathrm{SM}_i})$$

This key exchange process is also applicable during the OBU registration phase. Therefore, it is necessary to prove the correctness of Q'_v and R', which is shown as follows:

$$\begin{aligned}
Q'_\nu &= V_1 + x_\nu V_2 + \cdots + x_\nu^{k-1} (\mathrm{mod} q) V_k \\
&= \sum_{l=1}^{k} u_{l1} Q_l + x_\nu \left(\sum_{l=1}^{k} u_{l2} Q_l \right) + \cdots + x_\nu^{k-1} \left(\sum_{l=1}^{k} u_{lk} Q_l \right) \\
&= \left(\sum_{l=1}^{k} u_{1l} x_\nu^{l-1} \right) Q_1 + \left(\sum_{l=1}^{k} u_{2l} x_\nu^{l-1} \right) Q_2 + \cdots \\
&+ \left(\sum_{l=1}^{k} u_{\nu l} x_\nu^{l-1} \right) Q_\nu + \cdots + \left(\sum_{l=1}^{k} u_{kl} x_\nu^{l-1} \right) Q_k \\
&= f_1(x_\nu) Q_1 + f_2(x_\nu) Q_2 + \cdots + f_\nu(x_\nu) Q_\nu \\
&= Q_\nu \\
R' &= \left(\mathrm{SK}^1_{\mathrm{SM}_\nu} + \mathrm{SK}^2_{\mathrm{SM}_\nu} \right)^{-1} Q'_\nu \\
&= \left(\mathrm{SK}^1_{\mathrm{SM}_\nu} + \mathrm{SK}^2_{\mathrm{SM}_\nu} \right)^{-1} r_1 \left(\mathrm{PK}^1_{\mathrm{SM}_\nu} + \mathrm{PK}^2_{\mathrm{SM}_\nu} + h_\nu \mathrm{PK}_{\mathrm{AD}} \right) \\
&= \left(\mathrm{SK}^1_{\mathrm{SM}_\nu} + \mathrm{SK}^2_{\mathrm{SM}_\nu} \right)^{-1} r_1 \Big(\mathrm{SK}^1_{\mathrm{SM}_\nu} + d_\nu + h_\nu \mathrm{SK}_{\mathrm{AD}} \Big) P \\
&= \left(\mathrm{SK}^1_{\mathrm{SM}_\nu} + \mathrm{SK}^2_{\mathrm{SM}_\nu} \right)^{-1} r_1 \Big(\mathrm{SK}^1_{\mathrm{SM}_\nu} + \mathrm{SK}^2_{\mathrm{SM}_\nu} \Big) P \\
&= r_1 P \\
&= R
\end{aligned}$$

5.2 Security Analysis

Theorem 1. *The proposed BTCAA scheme ensures the anonymity of vehicle identities during the authentication process, such that no entity other than the AD can trace the real identity of the vehicle OBU_i.*

Proof. During the registration phase, the AD computes the vehicle's pseudonymous identifier sequence PID_j based on randomly generated parameters x and the generator P, together with the sequence number j and hash function H_2, as follows:

$$PID_j = H_2(j \cdot Y) \oplus OBU_i$$

The generation of the anonymous identifier depends on the restoration parameter $Y = x \cdot P$ and the real identity OBU_i, and the variation of the sequence number j ensures that each generated PID_j is different. Due to the unpredictability of the random parameter x and the confidentiality of the restoration parameter Y, any entity other than the AD cannot obtain Y or OBU_i, and even if they obtain PID_j, they cannot reverse-engineer OBU_i.

The one-wayness of the hash function H_2 and the randomness of the parameter x further enhance the security of the anonymous identifier, ensuring that different authentication requests' PID_j cannot be linked. Even if an attacker observes multiple authentication interactions, they cannot deduce the real identity of the vehicle or trace its behavior. Identity anonymity is fully preserved in this mechanism. □

Theorem 2. *The proposed BTCAA scheme enables the recovery of the real identity of the authentication initiator, thereby ensuring the traceability of anonymous vehicle identities.*

Proof. In the proposed BTCAA scheme, the anonymous identifier used by the OBU in the authentication process is generated by the AD during the registration phase. The real identity OBU_i and the restoration parameter Y are bound together as (OBU_i, Y) and stored locally. When the identifier is returned, it is encrypted using the shared key between the OBU and AD. When the SM questions the real identity of the authentication party, the recovered message $j, PID_{i,j}$ can be submitted to the AD. Upon receiving the message, the AD verifies it by checking the local record of (OBU_i, Y) and computes ${OBU_i}' = PID_{i,j} \oplus H_2(j \cdot Y)$. If any record satisfies ${OBU_i}' = OBU_i$, the real identity of the authentication initiator can be confirmed, and the traceability is completed.

Because the calculation of H_2 involves the combination of j and Y, where Y is a parameter strongly bound with OBU_i during registration, no entity other than the AD and the traced OBU can directly deduce Y or OBU_i. Thus, the traceability of the real identity is both secure and unique. By ensuring the integrity of key negotiation and identifier binding, this scheme can effectively trace anonymous identities. □

Theorem 3. *The proposed BTCAA scheme ensures that no entity other than the cross-domain authentication initiator and the receiving SM can decrypt the transmitted messages, thereby ensuring the confidentiality of the messages during authentication.*

Proof. During the registration phase, the OBU and AD generate a shared key based on elliptic curve cryptography. Suppose the OBU's private key is SK_{OBU} and its public key is $PK_{OBU} = SK_{OBU} \cdot P$; the AD's private key is SK_{AD} and its public key is $PK_{AD} = SK_{AD} \cdot P$. The shared key between the two parties is computed as:

$$K_{sharedOBU} = SK_{OBU} \cdot PK_{AD} = SK_{OBU} \cdot (SK_{AD} \cdot P).$$

This shared key is used to symmetrically encrypt the registration message M, resulting in the ciphertext:

$$C = \mathrm{Enc}_{K_{sharedOBU}}(M).$$

If the ECDHA holds, the adversary cannot compute the shared key from the known values of PK_{OBU} and PK_{AD}, nor can they decrypt the ciphertext C. Therefore, the confidentiality of messages during the registration phase is guaranteed.

During the authentication phase, the SM must use its ID hash identifier along with the received ciphertext $\delta = (V_1, V_2, \ldots, V_k, C)$ to derive the correct Q'_ν in order to obtain subsequent authentication parameters. Although the IDs of all domain SMs are public, meaning an adversary can easily obtain Q'_ν, the adversary still needs the corresponding private key of the SM to retrieve R' and decrypt the authentication parameters. Therefore, the confidentiality of messages during both the registration and authentication phases is effectively ensured. □

Theorem 4. *The proposed BTCAA scheme ensures that the adversary cannot alter the messages during the registration and authentication phases, thereby effectively guaranteeing the integrity of the messages.*

Proof. During the registration phase, the first part of the public key and identity identifier for both the OBU and SM (with the OBU's identity identifier encrypted) are transmitted to the AD. If an attacker tries to intercept the communication and replace the real public key with a forged one to interact with the AD, the attack will be difficult to execute. The reason is that the attacker would need to derive the AD's private key SK_{AD} using the generator P and AD's public key PK_{AD} in order to generate the correct shared key and encrypt the returned information. However, if the ECDLA holds, the attacker cannot reverse-engineer SK_{AD} from

P and PK_{AD}. Furthermore, the attacker cannot generate the correct second private key, as this depends on the AD's private key. This discrepancy would be easily detected during the SM or OBU key validity check. Therefore, the integrity of messages during the registration phase is effectively ensured.

During the authentication phase, if the transmitted message $\delta = (V_1, V_2, \ldots, V_k, C)$ is tampered with, the SMs would be unable to compute the correct parameters from the ciphertext, resulting in authentication failure. Thus, tampered messages could be effectively detected and rejected. □

Theorem 5. *The proposed BTCAA scheme can defend against replay attacks, ensuring the security of the authentication process.*

Proof. Each authentication request embeds a dynamic timestamp t_j, tightly bound to: $c_{j,2} \cdot P = c_{j,1} + H_1(c_{j,1}, t_j, PID_{i,j}) \cdot (PK^1_{OBU_i} + PK^2_{OBU_i} + H_0(PID_{i,j}, PK^2_{OBU_i}, PK^1_{OBU_i}) \cdot PK_{AD})$. Replayed messages with expired t_j are detected during signature verification. Adversaries cannot forge valid $c_{j,2}$ for modified t_j if the ECDLA holds. □

6 Theoretical Analysis and Experimental Comparison

In this section, we compare the schemes proposed by Yao et al. [24] and Cui et al. [25] with the BTCAA scheme.

6.1 Efficiency Analysis

Table 1: Symbols definition

Symbol	Description
T_{mul}	Time of calculating multiplication operations on the large element.
T_{div}	Time of calculating division operations on the large element.
T_{pm}	Time of calculating a scalar point multiplication of ECC.
T_{pa}	Time of calculating a point addition operation of ECC.
T_{ca}	Communication time with the authentication server in the domain
T_{cb}	Communication time with the blockchain
T_{ec}	The execution time of the smart contract.

Table 2: Comparison of efficiency

Scheme	Sender computation overhead	Receiver computational overhead
Cui et al. [25]	$nT_{mul} + nT_{pm}$	$nT_{mul} + nT_{pm} + 2nT_{ca} + 2nT_{cb} + nT_{ec}$
Yao et al. [24]	$(n+4)T_{pm} + (n+2)T_{pa} + T_{div} + nT_{mul}$	$(n+3)T_{pm} + (n+2)T_{pa} + nT_{mul}$
BTCAA	$(n+4)T_{pm} + (n+2)T_{pa} + nT_{mul}$	$(n+3)T_{pm} + (n+2)T_{pa} + nT_{mul}$

To represent the theoretical computational efficiency of each scheme, symbols for various operations are defined in Table 1. Table 2 lists the theoretical time costs for the authentication initiator and verifier for n domains. The scheme by Cui et al. [25] performs better in the authentication initiation phase due to offloading computations to edge devices. However, during

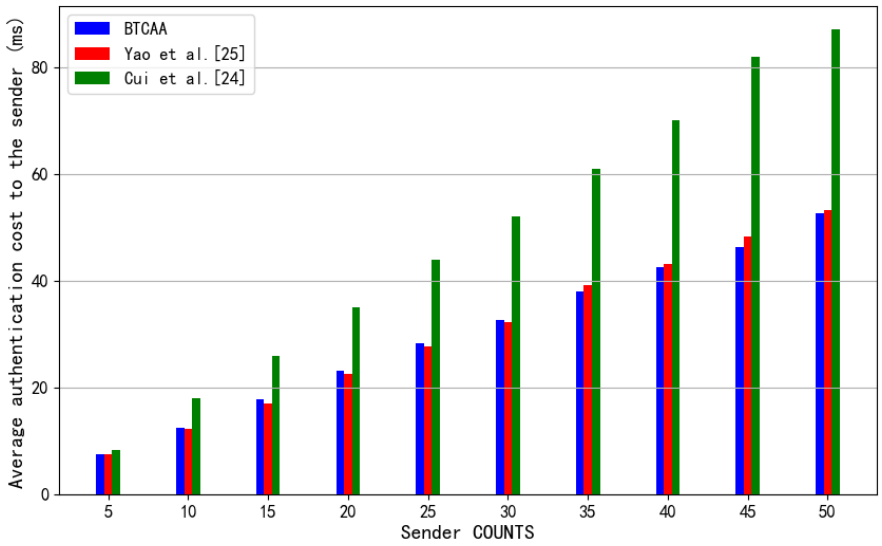


Figure 2: Average authentication cost for the receiver with varying number of senders

the authentication reception phase, reliance on the authentication server and blockchain may cause system overload, affecting efficiency.

In contrast, Yao et al. [24] and our BTCAA scheme avoid reliance on external entities throughout the process. Regardless of the number of domains n, the authentication initiator only needs one computation. The BTCAA scheme adds identity restoration without increasing overhead and avoids large division operations, reducing the initiator's computational burden.

Table 2 shows that Cui et al.'s scheme [25] has lower sender computational overhead but higher receiver communication costs due to blockchain and server reliance. In comparison, Yao et al. [24] and BTCAA schemes maintain balanced performance and optimize the computation process, ensuring efficient authentication with minimal additional overhead.

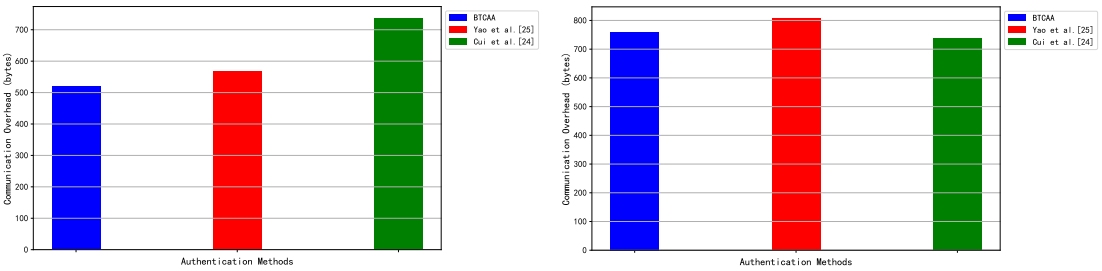
6.2 Experimental Comparison

Table 3: Experiment environments

Environment		Details
Hardware	CPU	12th Gen Intel(R) Core(TM) i5-12500
	Memory	16GB
Software	Operating system	Microsoft Windows 11
	Programming language	Python in Pycharm 2024.1.2
	Library	Pairing Based Cryptography

BTCAA was experimented with using the ECDSA library https://github.com/ecdsa/python-ecdsa and the PyCryptodome library https://www.pycryptodome.org/. The detailed information about the hardware and software environments are summarized in Table 3. In the experiment, the elliptic curve SECP256k1 (a standard elliptic curve) was used, where the parameter q is a 256-bit prime, and the order of the generator G is also 256 bits.

The time overhead of the receiver authentication process is shown in Figure 2. As seen, within the same authentication batch, the authentication time gradually increases as the number of senders grows. Compared to the scheme by Cui et al. [25], the authentication time increases significantly with the increasing number of senders, mainly due to the reliance on multi-party collaboration and blockchain smart contract operations during the authentication phase. This causes an increase in system burden as the number of senders increases. While the authentication time is generally comparable between the scheme by Yao et al. [24] and this approach, the latter additionally supports the subsequent restoration of the real identity, which significantly enhances the system's security — an attribute not provided by the other two schemes.



(a) Communication overhead with 5 SMs (b) Communication overhead with 10 SMs

Figure 3: Comparison of communication overhead with 5 and 10 SMs.

The communication overhead during the authentication process is shown in Figure 3. In Figure 3a, we set the number of domains to be crossed to 5, and only the communication overhead of a single transmission is calculated. In this scheme, the message sent by the OBU to the SM is $(V_1, V_2, \ldots, V_5, C)$, where the length of $V_1, V_2, \ldots, V_5$ is 240 bytes, and C is 279 bytes. The communication overhead is calculated to be 519 bytes in this case. Compared to Yao et al.'s scheme [24], the communication overhead increases by 48 bytes under the same conditions, due to the need to send an additional point element and its serialized x and y coordinates. In contrast to the scheme by Cui et al. [25], which depends on other parties' involvement, the communication overhead reaches 737 bytes. In Figure 3b, we set the number of cross-domains to 10. As the number of domains increases, the number of V_k parameters also increases, while the size of C remains unchanged. The total size increases from 519 bytes to 759 bytes as V_k grows from 5 to 10. While this is slightly higher than the communication overhead in Cui et al.'s scheme [25], it still outperforms Yao et al.'s scheme [24]. Overall, despite the increase in communication overhead with more cross-domain scenarios, BTCAA maintains good communication efficiency compared to other existing schemes.

7 Conclusion

This paper designed a blockchain-based cross-domain anonymous authentication mechanism for VFS, named BTCAA. The BTCAA mechanism achieves flexible and efficient verification through dynamically generated parameters, significantly reducing overhead with a single communication round. BTCAA protects user privacy with dynamic pseudonyms and resists tampering attacks via blockchain's decentralized nature. Crucially, it ensures accountability by enabling the tracing of a vehicle's true identity in case of disputes. Security analysis and simulations demonstrated the BTCAA scheme's feasibility and effectiveness.

Acknowledgement

This article is supported in part by the Guangxi Natural Science Foundation (2025GXNSFGA069004, 2025GXNSFAA069678), the National Natural Science Foundation of China (62162017, 62372067), Zhejiang Provincial Natural Science Foundation of China (LZ23F020012), the Guangxi Young Teachers' Basic Ability Improvement Program (2024KY0224), and the Guangxi Key Laboratory of Digital Infrastructure (GXDIOP2023006).

References

[1] Botta Alessio, Walter De Donato, Valerio Persico, and Antonio Pescapé. On the integration of cloud computing and internet of things. *Proc. Future internet of things and cloud (FiCloud)*, pages 23–30, 2014.

[2] Mohammad Shojafar, Nicola Cordeschi, and Enzo Baccarelli. Energy-efficient adaptive resource management for real-time vehicular cloud services. *IEEE Transactions on Cloud computing*, 7(1):196–209, 2016.

[3] Xueshi Hou, Yong Li, Min Chen, Di Wu, Depeng Jin, and Sheng Chen. Vehicular fog computing: A viewpoint of vehicles as the infrastructures. *IEEE Transactions on Vehicular Technology*, 65(6):3860–3873, 2016.

[4] Jie Lin, Wei Yu, Nan Zhang, Xinyu Yang, Hanlin Zhang, and Wei Zhao. A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE internet of things journal*, 4(5):1125–1142, 2017.

[5] Wanjun Xiong, Yujue Wang, and Yongzhuang Wei. Ntru-cls: Efficient quantum-resistant ntru lattice-based certificateless signature scheme for vanets. *Computer Networks*, 256:110885, 2025.

[6] Zhaojun Lu, Gang Qu, and Zhenglin Liu. A survey on recent advances in vehicular network security, trust, and privacy. *IEEE Transactions on Intelligent Transportation Systems*, 20(2):760–776, 2019.

[7] Baodong Wen, Yujue Wang, Yong Ding, Haibin Zheng, Bo Qin, and Changsong Yang. Security and privacy protection technologies in securing blockchain applications. *Information Sciences*, 645:119322, 2023.

[8] Hai Zhang and Feng Zhao. Cross-domain identity authentication scheme based on blockchain and pki system. *High-Confidence Computing*, 3(1):100096, 2023.

[9] Shengyu Duan and Gaole Sai. A secure authentication scheme based on differential public puf. In *Proceedings of the 19th ACM International Conference on Computing Frontiers*, pages 263–269, 2022.

[10] Yujue Wang, Yong Ding, Qianhong Wu, Yongzhuang Wei, Bo Qin, and Huiyong Wang. Privacy-preserving cloud-based road condition monitoring with source authentication in vanets. *IEEE Transactions on Information Forensics and Security*, 14(7):1779–1790, 2019.

[11] Lemei Da, Wei Hu, Yujue Wang, Yong Ding, Beibei Li, and Dan Zhu. A blockchain-assisted road condition monitoring scheme with privacy-preserving for vanets. *IEEE Transactions on Vehicular Technology*, 73(12):19625–19640, 2024.

[12] Mochan Fan, Zhipeng Zhang, Zonghang Li, Gang Sun, Hongfang Yu, and Mohsen Guizani. Blockchain-based decentralized and lightweight anonymous authentication for federated learning. *IEEE Transactions on Vehicular Technology*, 72(9):12075–12086, 2023.

[13] Badiea Abdulkarem Mohammed, Mahmood A Al-Shareeda, Selvakumar Manickam, Zeyad Ghaleb Al-Mekhlafi, Abdulaziz M Alayba, and Amer A Sallam. Anaa-fog: A novel anonymous authentication scheme for 5g-enabled vehicular fog computing. *Mathematics*, 11(6):1446, 2023.

[14] Linsheng Yu, Mingxing He, Hongbin Liang, Ling Xiong, and Yang Liu. A blockchain-based authentication and authorization scheme for distributed mobile cloud computing services. *Sensors*,

23(3):1264, 2023.

[15] Witali Bartsch, Owen Millwood, and Elif Kavun. Zero knowledge registration of pki authentication for symbiotic security in fido iot. *Journal of Surveillance, Security and Safety*, 4:155–79, 12 2023.

[16] Zhihua Cui, Fei XUE, Shiqiang Zhang, Xingjuan Cai, Yang Cao, Wensheng Zhang, and Jinjun Chen. A hybrid blockchain-based identity authentication scheme for multi-wsn. *IEEE Transactions on Services Computing*, 13(2):241–251, 2020.

[17] Caifen Wang, Chao Liu, Shufen Niu, Li Chen, and Xu Wang. An authenticated key agreement protocol for cross-domain based on heterogeneous signcryption scheme. In *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 723–728. IEEE, 2017.

[18] Yizhong Liu, Andi Liu, Yu Xia, Bin Hu, Jianwei Liu, Qianhong Wu, and Prayag Tiwari. A blockchain-based cross-domain authentication management system for iot devices. *IEEE Transactions on Network Science and Engineering*, 11(1):115–127, 2024.

[19] Lingyan Xue, Haiping Huang, Fu Xiao, and Wenming Wang. A cross-domain authentication scheme based on cooperative blockchains functioning with revocation for medical consortiums. *IEEE Transactions on Network and Service Management*, 19(3):2409–2420, 2022.

[20] Chaosheng Feng, Bin Liu, Zhen Guo, Keping Yu, Zhiguang Qin, and Kim-Kwang Raymond Choo. Blockchain-based cross-domain authentication for intelligent 5g-enabled internet of drones. *IEEE Internet of Things Journal*, 9(8):6224–6238, 2021.

[21] Jie Deng, Lili Jiao, Lili Zhang, and Yongjin Ren. Research on cross-domain authentication scheme for v2g networks based on sm9 signature cryptography algorithm and consortium blockchain technology. In *International Conference on Intelligent Networking and Collaborative Systems*, pages 372–381. Springer, 2023.

[22] Ruibiao Chen, Fangxing Shu, Shuokang Huang, Lei Huang, Huafang Liu, Jin Liu, and Kai Lei. Bidm: a blockchain-enabled cross-domain identity management system. *Journal of Communications and Information Networks*, 6(1):44–58, 2021.

[23] Meng Shen, Huisen Liu, Liehuang Zhu, Ke Xu, Hongbo Yu, Xiaojiang Du, and Mohsen Guizani. Blockchain-assisted secure device authentication for cross-domain industrial iot. *IEEE Journal on Selected Areas in Communications*, 38(5):942–954, 2020.

[24] Yingying Yao, Xiaolin Chang, Jelena Mišić, Vojislav B Mišić, and Lin Li. Bla: Blockchain-assisted lightweight anonymous authentication for distributed vehicular fog services. *IEEE Internet of Things Journal*, 6(2):3775–3784, 2019.

[25] Jie Cui, Nan Liu, Qingyang Zhang, Debiao He, Chengjie Gu, and Hong Zhong. Efficient and anonymous cross-domain authentication for iiot based on blockchain. *IEEE Transactions on Network Science and Engineering*, 10(2):899–910, 2022.