A Robust and Verifiable Data Sharing Architecture Driven by Cloud-Edge Coordination

Mengjun Wang[1, 4], Chunbo Wang[1, 4], Chunhua Su[2], Xiaoqiang Di[1, 4], Nannan Xie[1, 4], Pengfei Hu[3], and Hui Qi[1, 4, *]
[1]School of Computer Science and Technology, Changchun University of Science and Technology, Changchun, Jilin, 130022, China
[2]Department of Computer Science and Engineering, Division of Computer Science, University of Aizu, Aizuwakamatsu, 965-8580, Japan
[3]School of Computer Science and Technology, Shandong University, Qingdao, Shandong, 266237, China
[4]Jilin Province Key Laboratory of Network and Information Security, Changchun University of Science and Technology, Changchun, Jilin, 130022, China

The Internet of Vehicles (IoV) generates massive real-time data involving both user privacy and driving safety. To overcome the single-point-of-failure and key abuse risks of centralized cloud storage, we propose a secure and verifiable data sharing scheme based on cloud–edge collaboration. The scheme leverages edge–cloud cooperation to improve data access efficiency, employs ciphertext-policy attribute-based encryption (CP-ABE) for fine-grained access control, and adopts a multi-authority architecture with a permissioned PBFT-based blockchain deployed on resource-rich regional servers to ensure consistent key distribution without burdening resource-constrained edge nodes. In addition, BLS short signatures are integrated to verify data integrity. Performance evaluation of the blockchain prototype demonstrates acceptable transaction latency and manageable communication overhead, confirming the practicality of key consensus in real-world IoV deployments.

***Index Terms*—Vehicle Information Security, Cloud–Edge Collaboration, Ciphertext-Policy Attribute-Based Encryption (CP-ABE), Key Consensus, Data Tamper-Resistance.**

I. Introduction

With the rapid development of intelligent transportation systems and the Internet of Vehicles (IoV), the volume of data exchanged between vehicles and their surrounding environments is steadily increasing. Due to the limited computing and storage capabilities of in-vehicle devices, many services still rely on remote cloud platforms. However, traditional centralized cloud architectures exhibit significant drawbacks in practical deployments: on the one hand, if the cloud center suffers from attacks or failures, all vehicular data can be compromised, leading to severe privacy leakage risks [1]; on the other hand, cross-region access to cloud platforms often results in high latency, which fails to meet the stringent real-time and reliability requirements of IoV [2]. To address these issues, increasing attention has been paid to cloud–edge collaborative architectures, in which part of the storage and computation tasks are offloaded to edge nodes located near vehicles. This paradigm effectively reduces transmission overhead and improves data processing speed and responsiveness during vehicle operation [3].Beyond performance bottlenecks, data security remains another critical challenge for IoV. Vehicular data often contains sensitive information such as driving trajectories and identity credentials. Once disclosed, such data can raise concerns about not only privacy, but also potential threats to traffic safety [4]. Therefore, data-sharing mechanisms must simultaneously ensure both usability and fine-grained access control, allowing only authorized users to retrieve information. Attribute-Based Encryption (ABE) has been widely applied in vehicular environments due to its ability to enforce flexible access policies. In particular, Ciphertext-Policy ABE (CP-ABE) offers high flexibility in vehicular data sharing. However, traditional single-authority models suffer from single points of failure and remain vulnerable to key abuse and collusion attacks [5].Despite these advancements, existing CP-ABE-based data sharing schemes for IoV still face critical challenges that hinder their practical deployment. First, most schemes rely on centralized cloud storage, leading to high cross-region access latency and single-point-of-failure risks that are particularly detrimental in latency-sensitive and safety-critical vehicular environments. Second, although multi-authority mechanisms mitigate single-point issues in key management, they often lack robust consensus protocols to prevent key abuse or inconsistent key generation by malicious or compromised authorities, leaving systems vulnerable to collusion and misuse attacks. Third, integrating blockchain for enhanced trustworthiness typically introduces excessive computational and communication overhead, especially when deployed on resource-constrained edge devices. Finally, few schemes provide lightweight yet effective integrity verification for tamper-prone transmissions in dynamic IoV networks while simultaneously achieving fine-grained access control, low latency via cloud-edge collaboration, and scalable decentralized key management. Consequently, designing a secure, efficient, and scalable data sharing framework that jointly addresses these intertwined challenges—low-latency storage, trustworthy multi-authority key distribution, lightweight integrity protection, and practical consensus overhead—remains an open and critical

Manuscript received June 15, 2025; revised September 5, 2025. Corresponding author: Hui Qi (email: qihui@cust.edu.cn).

research gap in IoV.To tackle these challenges, this paper proposes a security-enhanced and application-oriented data-sharing mechanism tailored for IoV. Specifically, multiple distributed authorization servers are introduced to collaboratively manage keys, while a permissioned PBFT-based blockchain is used to record operations and maintain consistency. This design not only ensures transparency and trust, but also improves system scalability and resilience against attacks. Moreover, in the data transmission phase, a lightweight signature verification mechanism is integrated to provide fast integrity checks for resource-constrained vehicular devices, thereby ensuring reliable communication in highly dynamic network environments. The Authorization Servers (ASs) responsible for blockchain and PBFT consensus are designed to be limited in number (typically 10–20 per urban area or administrative region) and deployed as resource-rich regional entities (e.g., traffic management centers or dedicated cloudlets), distinct from the lightweight edge nodes (ENs) that only handle ciphertext storage and forwarding. The PBFT consensus is invoked only during infrequent attribute key generation and distribution, rather than per-data-access, ensuring minimal impact on real-time data sharing performance.The main contributions of this work are summarized as follows:

- **Low-latency data services for IoV.** By introducing edge nodes close to vehicles, a cloud–edge collaborative processing model is established to reduce frequent interactions with remote clouds. Under large-scale service requests, this design significantly decreases communication latency and improves the timeliness of data acquisition and service response in vehicular environments.
- **Decentralized key management mechanism.** To overcome the vulnerability and performance bottlenecks of centralized authorities, a multi-authority key management framework is proposed, in which multiple regional servers collaboratively participate in key distribution. Blockchain is leveraged to achieve traceability and transparency of key operations, thereby enhancing the security and scalability of vehicular data sharing in real-world deployments.
- **Lightweight integrity verification for constrained devices.** Considering the limited computational resources of on-board units and certain IoT nodes, an efficient signature scheme is employed to enable fast data integrity verification during sharing. Experimental results demonstrate that the proposed method achieves strong security guarantees while maintaining low computational and storage overhead, making it suitable for practical IoV applications.

II. Related Work

In data sharing for the Internet of Vehicles (IoV), achieving secure and fine-grained access control remains a fundamental challenge. Attribute-Based Encryption (ABE), first proposed by Goyal et al. [7], provides the theoretical basis for attribute-oriented access control. To enhance policy expressiveness, Ciphertext-Policy ABE (CP-ABE) was subsequently explored. Bethencourt et al. [6] introduced the first CP-ABE construction, Cheung et al. [8] adopted AND-gate–based policies, and Lewko et al. [9] further incorporated Linear Secret Sharing Schemes (LSSS) to improve flexibility. Wang et al. [10] extended the traditional access tree with weighted attributes, but such schemes are inefficient for resource-constrained IoV environments.

Early CP-ABE systems rely on a single authority, which may cause single-point failures and vulnerability to key abuse. Liu et al. [11] applied CP-ABE to IoV access control, ensuring that only authorized vehicles can obtain data; however, its single authorization server risks becoming a performance bottleneck.

To overcome the drawbacks of single-authority models, researchers proposed multi-authority CP-ABE schemes. Zhang et al. [16] designed a decentralized multi-authority architecture with full policy hiding, although the ciphertext overhead remains large. Horng et al. [17] presented a multi-authority IoT data sharing system suitable for cloud environments but with high response latency. Zhao et al. [29] and Liu et al. [18] further developed multi-authority schemes; however, they did not incorporate blockchain [19], [20] and therefore cannot resist key misuse attacks [21], [22].

Recent studies focus on improving the efficiency and verifiability of CP-ABE [12], [13], [14], [15]. To address real-time demands in IoV, several works combine CP-ABE with blockchain and support online/offline encryption, verifiable outsourced decryption, or revocation mechanisms [15], [23], [24]. However, these schemes still depend on centralized cloud storage, resulting in high latency and single-point failure risks [23], [24], [25]. Some researchers offload computation to RSUs or edge nodes [18], [22], [27], but the security and coordination between cloud and edge remain underexplored.

Further advances include multi-authority CP-ABE combined with emerging technologies. Xie et al. [28] integrated blockchain and elliptic curve cryptography; Zhao et al. [29] introduced online/offline multi-authority CP-ABE; Wang et al. [30] addressed cross-domain data sharing; Li and Roy [31], [32] improved efficiency and hierarchical administration; and Duan et al. [33] proposed multi-authority proxy re-encryption for cross-blockchain sharing. Although these approaches mitigate collusion attacks and offer enhanced security, key management remains complex and cloud–edge collaboration is not exploited adequately.

In summary, existing CP-ABE-based IoV schemes still suffer from single-point failures, key abuse risks, and high latency, and few fully integrate cloud–edge collaboration with scalable decentralized key management. Designing a secure, efficient, and adaptable data sharing framework that simultaneously achieves low-latency storage, trustworthy multi-authority key distribution via robust consensus, and lightweight integrity verification remains an open research challenge. Although existing multi-authority schemes often overlook scalability, edge resource constraints, and blockchain-specific metrics—leaving them vulnerable to key abuse and inconsistent generation by malicious authorities [21], [22]—our design addresses these gaps by employing a PBFT-based authorized blockchain on a limited number of resource-sufficient regional servers (typically 10–20 per region) and restricting consensus to infrequent key issuance. This eliminates Gas costs inherent in

public blockchains, confines overhead to well-resourced nodes, and maintains efficiency without overburdening lightweight edge or vehicular devices—a critical limitation inadequately resolved in prior works.

III. Preliminaries

A. Bilinear Maps

Definition (Bilinear Mapping). Let G and G_T be two cyclic multiplicative groups of prime order p, and let g be a generator of G. A bilinear map is a function

that satisfies the following properties:

1) Bilinearity:

For all $g_1, g_2 \in G$ and $a, b \in \mathbb{Z}_p^*$,

$$e\left(g_1^a, g_2^b\right) = e(g_1, g_2)^{ab} \tag{1}$$

2) Non-degeneracy:

There exist $g_1, g_2 \in G$ such that

$$e(g_1, g_2) \neq 1 \tag{2}$$

3) Computability:

For all $g_1, g_2 \in G$, there exists an efficient algorithm to compute $e(g_1, g_2)$. In addition, the bilinear map has the following useful properties:

1) $\forall g_1, g_2 \in G$

$$e(g_1 g_2, g) = e(g_1, g)e(g_2, g) \tag{3}$$

2) $\forall g_1, g_2 \in G$, $\forall a, b \in Z_p^*$

$$e\left(g_1^a, g_2^b\right) = e(g_1, g_2)^{ab} = e\left(g_1^b, g_2^a\right) \tag{4}$$

B. Linear Secret Sharing Scheme, LSSS

Definition (Linear Secret Sharing Scheme (LSSS)). Let $S = S_1, S_2, \ldots, S_n$ denote the set of user attributes, and let (M, ρ) represent the access policy defined by the Data Owner (DO), where M is an $l \times n$ access matrix, and ρ is a mapping function that maps each row of M to an attribute $S_i \in S$. A Linear Secret Sharing Scheme (LSSS) consists of the following algorithms:

1) Secret Sharing $((M, \rho), s)$*:*

Given a secret $s \in \mathbb{Z}_p$ to be shared and the attribute set S, the algorithm generates shares of s. Let $v = (s, v_1, v_2, \ldots, v_n)$ be a random vector, where $v_1, v_2, \ldots, v_n \in \mathbb{Z}_p$ are randomly chosen. The share corresponding to attribute $\rho(i)$ is computed as

$$\lambda_i = M_i \cdot v \tag{5}$$

where M_i is the i-th row of M.

2) Linear Reconstruction $(\lambda_i, \ldots, \lambda_l, (M, \rho))$*:*

The secret s can be reconstructed linearly from a set of shares. Let $A \subseteq S$ be any authorized set of attributes, and define $I = i : \rho(i) \in A \subseteq 1, 2, \ldots, l$. There exist constants $c_i \mid i \in I$ such that

$$\sum_{i \in I} c_i M_i = (1, 0, \ldots, 0) \tag{6}$$

Consequently, the secret can be recovered as

$$\sum_{i \in I} c_i \lambda_i = s. \tag{7}$$

C. Practical Byzantine Fault Tolerance, PBFT

In blockchain-based distributed systems, the consensus speed among nodes directly affects system efficiency, while consensus algorithms ensure data consistency and enhance storage performance and scalability. Common consensus algorithms include Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS), and Practical Byzantine Fault Tolerance (PBFT) . In vehicular network scenarios, where nodes are numerous and information exchange is complex, high requirements are imposed on data security, low latency, and high throughput. Compared with other algorithms, PBFT offers higher consensus efficiency and stronger real-time performance, making it suitable for high-frequency transactions and dynamic environments.

PBFT can tolerate up to $(n-1)/3$ faulty or malicious nodes while still reaching correct consensus. In the algorithm, all replica nodes participate in view management, where each view is uniquely identified by a number v. The view number v is initialized at system setup and incremented upon each view change. The primary node is selected according to

$$p = v \bmod |R| \tag{8}$$

where $|R|$ denotes the total number of replica nodes; the remaining nodes serve as backups. If the primary node fails, a view change is triggered, updating the view number and re-electing a primary. Although PBFT introduces O(n²) communication complexity, in our scheme it incurs no Gas consumption (characteristic of permissioned consensus) and is applied exclusively to infrequent attribute key generation (typically once per user registration or attribute update). The number of participating ASs is strictly limited (typically 10–20 per region, far smaller than vehicle or edge node counts) and deployed on resource-rich servers. As validated in the performance evaluation, consensus delay remains practical even at 20 ASs, confirming feasibility without burdening edge devices.

The consensus process proceeds as follows: a client sends a request to the primary node, which broadcasts a pre-prepare message. Backup nodes then generate prepare and commit messages sequentially. Once a node receives a sufficient number of commit messages, it executes the requested operation. The client confirms the completion of its request upon receiving responses from more than $(n-1)/3$ nodes, thereby ensuring fault tolerance and data consistency.

D. Boneh–Lynn–Shacham Signature

BLS (Boneh-Lynn-Shacham) Signature Scheme is a digital signature algorithm based on bilinear pairings, featuring simplicity, signature aggregation, and uniqueness. The construction of BLS signatures consists of four main components: initialization, key generation, signing, and verification, described as follows:

1) Initialization:

Let G_1 and G_2 be cyclic multiplicative groups of prime order p, with generators g_1 and g_2, respectively. Let $e : G_1 \times$

$G_2 \to G_T$ denote a bilinear map, and let $H : 0,1^* \to G_1$ be a hash function. The public parameters are

$$(G_1, G_2, G_T, e, g_1, g_2, p, H)$$

2) Key Generation:

Select a random $x \in \mathbb{Z}_p^*$ as the private key. The corresponding public key is

$$v = g_2^x \in G_2 \tag{9}$$

Thus, the key pair is (x, v).

3) Signing:

For a message m, map it to a point in the group as $h = H(m)$. The signature is then generated using the private key:

$$\delta = h^x \tag{10}$$

4) Verification:

Given the signature δ and the public key v, the verifier checks the equality

$$e(\delta, g_2) \stackrel{?}{=} e(h, v) \tag{11}$$

which can be expanded as

$$e(\delta, g_2) = e(h^x, g_2) = e(h, g_2^x) = e(h, v) \tag{12}$$

Advantages of BLS Signatures:

1) Simplicity:

Compared to traditional signature schemes, BLS signatures are computationally simpler. The signer only needs to perform an exponentiation on the hash of the message, resulting in lower computational complexity.

2) Aggregatable Signatures

BLS signatures support aggregation, allowing multiple signatures to be combined into a single signature for verification. This property is particularly useful when verifying multiple signatures, significantly reducing storage and verification time.

3) Uniqueness:

For the same message, the BLS signature is unique. Even if the same message is signed multiple times, the resulting signature remains identical, enhancing verifiability and tamper resistance.

E. Decisional q-Parallel Bilinear Diffie-Hellman Exponent Assumption

Definition (Decisional q-Parallel Bilinear Diffie-Hellman Exponent (BDHE) Assumption). Let G and G_T be two cyclic multiplicative groups of prime order p, with generator $g \in G$ and bilinear map $e : G \times G \to G_T$. The decisional q-parallel BDHE problem is defined as follows: Given a tuple

$$\vec{y} = \left(g,\ g^a,\ \ldots,\ g^{a^q},\ g^b,\ \{g^{a^i b^j / a^k}\}_{\substack{1 \le i,k \le q \\ 1 \le j \le q \\ k \ne i}}\right) \in G^{O(q^2)} \tag{13}$$

and an element $T \in G_T$, decide whether $T = e(g,g)^{a^{q+1}b}$ or T is a random element in G_T. The decisional q-parallel BDHE assumption holds if no probabilistic polynomial-time (PPT) adversary can distinguish the two cases with non-negligible advantage. This assumption is widely used in proving the security of attribute-based encryption schemes in the standard model.

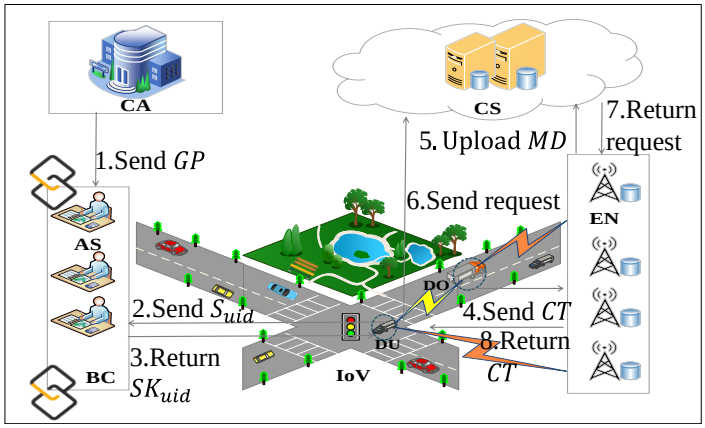


Fig. 1: Data Sharing Scenario

IV. Problem Formulation

A. System Model

This paper develops a cloud-edge collaborative data sharing system for the Internet of Vehicles, in which the main notations are summarized in Table I . The system involves seven types of entities: the Central Authority (CA), which is responsible for parameter generation and registration; the Authorization Server (AS), which authenticates identities, generates and distributes attribute keys, and ensures consistency among ASs through blockchain and the PBFT protocol; the Data Owner (DO), which collects and encrypts vehicular data before uploading it to the Edge Node (EN); the Data User (DU), which verifies signatures and decrypts data using attribute keys; the Cloud Storage (CS), which stores metadata and enforces access control; the EN, which reduces transmission latency; and the Blockchain (BC), which guarantees trustworthy collaboration among ASs.

As illustrated in Fig. 1 , the Central Authority (CA) initializes the system by generating the global public parameters GP and completing the registration of the Authorization Servers (ASs) and Data Users (DUs), issuing each AS and DU an identity identifier aid and uid, respectively. When a DU intends to obtain its attribute secret key SK_{uid}, it submits the attribute set S_{uid} together with the identifier uid to the AS. Acting as a blockchain node, the AS employs the PBFT protocol [34], [35] to reach consensus among all ASs on generating SK_{uid}, and then returns SK_{uid} to the DU.

During driving, the Data Owner (DO) collects vehicular data and defines an access policy (M, ρ) according to the attributes of DUs. The original data m is first encrypted with the AES algorithm to produce the ciphertext C_t. The symmetric key κ is then encrypted using the CP-ABE algorithm to obtain the key ciphertext C_κ. Afterwards, the DO applies the BLS short signature algorithm to sign the ciphertext set CT (consisting of C_t and C_κ), producing a signature σ_{CT}. Finally, the ciphertext set CT and the signature σ_{CT} are uploaded to the Edge Node (EN) for storage, while the EN generates the metadata MD and uploads it to the Cloud Storage (CS).

When a Data User (DU) intends to access the encrypted data, it sends its attribute set S_{uid} to the Cloud Storage (CS). The CS verifies whether S_{uid} satisfies the access policy

defined by the Data Owner (DO). If the condition holds, the CS retrieves the storage location identifier ID of the requested data from the metadata MD stored at the Edge Node (EN), and then sends a request for the corresponding ciphertext set CT to the EN with identifier ID. Upon receiving the request from the CS, the EN locates the ciphertext set CT and its signature σ_{CT} according to the ciphertext index set d, and returns them to the DU.

After obtaining CT and σ_{CT}, the DU first verifies the validity of the signature σ_{CT}. If the verification succeeds, the DU uses its attribute secret key SK_{uid} to decrypt the key ciphertext C_κ, thereby recovering the symmetric key κ. Finally, the DU employs κ to decrypt the ciphertext C_t and obtain the original data m. It is emphasized that the Authorization Servers (ASs) running the blockchain and PBFT protocol are resource-rich entities (e.g., deployed at regional traffic authorities or dedicated servers), and do not coincide with the resource-constrained edge nodes (ENs, such as roadside units). The ENs are only responsible for low-overhead ciphertext storage and proximity-based retrieval, while the computationally intensive consensus process occurs solely among ASs during user attribute key issuance.

B. Threat Model

The proposed scheme employs Ciphertext-Policy Attribute-Based Encryption (CP-ABE) to achieve fine-grained access control over shared data in the Internet of Vehicles (IoV). To ensure secure data sharing under cloud storage, the scheme considers the following potential security threats:

1) Data Privacy Protection

The vehicular data uploaded by vehicles often contains sensitive information. An adversary may attempt to infer both the content of the data and the attributes of users. Hence, the scheme must guarantee that only legitimate receivers are able to access the plaintext data.

2) Resistance to AS Collusion Attacks

Semi-trusted Authorization Servers (ASs) honestly generate attribute secret keys SK_{uid}, but they may also be curious about the data uploaded by Data Owners (DOs). If multiple ASs collude to obtain all attribute keys of a DU, the confidentiality of the encrypted data may be compromised. Therefore, the scheme must be resilient against collusion among ASs.

3) Resistance to DU Collusion Attacks

In certain cases, the attribute set of a single DU may not satisfy the access policy. However, by combining their attributes, multiple DUs might collectively satisfy the access structure and gain unauthorized access to sensitive data. To prevent such data leakage, the scheme must be secure against collusion among DUs.

4) Resistance to Key Abuse Attacks

Since ASs are not fully trusted, there exist potential risks of key abuse. For example, colluding ASs might generate inconsistent intermediate keys for the same DU, undermining both the uniqueness and security of SK_{uid}. Thus, the scheme must effectively defend against the risk of key abuse caused by colluding ASs.

5) Resistance to EN Tampering and Forgery Attacks

In the proposed system model, the Edge Node (EN) is a semi-trusted entity that may forge ciphertexts. Furthermore, ciphertexts may be tampered with by external attackers during transmission, which could prevent DUs from obtaining the correct data. Therefore, it is essential to ensure ciphertext integrity during the data sharing process.

C. Security Model

The security model of the proposed scheme can be formalized as a game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. In this model, $\mathcal{A}$ is allowed to request any secret keys that cannot be used to decrypt the challenge ciphertext. By querying all keys inconsistent with the challenge ciphertext, $\mathcal{A}$ may attempt to bypass the security definition. We denote the set of all authorization servers as S_A. The detailed definition of the security game is as follows:

Initialization Phase: The challenger $\mathcal{C}$ executes the $\mathsf{CASetup}$ algorithm to generate the system parameters GP, which are then sent to $\mathcal{A}$. The adversary defines a subset $S'_A \subseteq S_A$ as the set of corrupted ASs. By issuing queries to $\mathsf{ASSetup}$ and KeyGen, $\mathcal{C}$ generates the corresponding public/private key pairs. The public keys of the uncorrupted ASs in $S_A - S'_A$ are then sent to $\mathcal{A}$.

Query Phase 1: By sending the identity aid of an uncorrupted AS, $\mathcal{A}$ can request a series of key queries from $\mathcal{C}$. The challenger executes the KeyGen algorithm and returns the appropriate keys to $\mathcal{A}$.

Challenge Phase: When $\mathcal{A}$ initiates the challenge, it submits two equal-length messages m_0 and m_1, together with a challenge access structure (M^*, ρ^*), which is not satisfied by any queried key set from Phase 1. The challenger flips a random coin $b \in 0, 1$ and encrypts m_b under(M^*, ρ^*), producing the challenge ciphertext CT^*, which is returned to $\mathcal{A}$.

Query Phase 2: As long as no attribute set satisfies the challenge access structure (M^*, ρ^*), $\mathcal{A}$ may continue to issue key queries as in Phase 1, or request key updates.

Guess Phase: Finally, $\mathcal{A}$ outputs a guess $b' \in 0, 1$ for the bit b. The advantage of $\mathcal{A}$ in this game is defined as:

$$\mathrm{Adv} = \Pr[b' = b] - \frac{1}{2} \tag{14}$$

Definition 1. The proposed scheme is secure if, for any polynomial-time adversary $\mathcal{A}$, its advantage in the above game is negligible.

Definition 2. The proposed scheme is collusion-resistant if no polynomial-time adversary $\mathcal{A}$ can decrypt a ciphertext by combining the public and private keys of different users, when each individual user alone cannot satisfy the access policy and thus cannot decrypt the data.

D. Algorithm Definition

1) $CASetup(\lambda) \rightarrow \{GP, aid, uid\}$:

The Central Authority (CA) executes this algorithm to initialize the system. Given the security parameter λ, it outputs the global public parameters GP. For each legitimate AS and DU, the CA generates a unique identity identifier aid and uid.

TABLE I: Meaning of Main Symbols in the Proposed Scheme

Symbol	Meaning	Symbol	Meaning
GP	System public parameters	MSK	Master secret key
PK_{aid}	Public key of attribute authority (AS)	SK_{aid}	Private key of attribute authority (AS)
PK_d	Public key of data owner (DO)	SK_d	Private key of data owner (DO)
aid	Identity of attribute authority (AS)	uid	Identity of data user (DU)
S_{uid}	Attribute set of data user (DU)	SK_{uid}	Attribute secret key of data user (DU)
C_t	Ciphertext	c_κ	Encrypted symmetric key
κ	Symmetric key	CT	Ciphertext set
MD	Metadata	σ_{CT}	Signature of ciphertext set
m	Original data	d	Index set
ID	Location identifier of ciphertext set	Len	Size of ciphertext

2) $ASSetup(GP) \rightarrow \{MSK, PK_{aid}, SK_{aid}\}$:

Each Authorization Server (AS) executes this algorithm by taking GP as input, and outputs the master secret key MSK, its public key PK_{aid}, and private key SK_{aid}.

3) $DOSetup(GP) \rightarrow \{PK_d, SK_d\}$;

The Data Owner (DO) executes this algorithm by taking GP as input, and outputs its public/private key pair (PK_d, SK_d).

4) $KeyGen(GP, MSK, SK_{aid}, S_{uid}, uid) \rightarrow \{SK_{uid}\}$;

The ASs execute this algorithm to generate the attribute secret key SK_{uid} for a DU under their management. The inputs include MSK, SK_{aid}, the attribute set S_{uid}, and the identifier uid. The PBFT protocol is employed to ensure consensus among ASs when generating SK_{uid}.

5) $Encrypt\,(GP, m, PK_{aid}, (M, \rho)) \rightarrow \{CT\}$:

This algorithm is executed by the DO. Given GP and PK_{aid}, the DO first encrypts the data m with AES to obtain the ciphertext C_t. Then, the symmetric key κ is encrypted using the CP-ABE algorithm under the access policy (M, ρ), producing the key ciphertext C_κ. The final ciphertext set CT consists of C_t, C_κ.

6) $Sign(GP, CT, SK_d) \rightarrow \{\sigma_{CT}\}$;

The DO executes this algorithm by taking GP, SK_d, and the ciphertext set CT as input. Using the BLS short signature algorithm, it generates a signature σ_{CT} on CT. Both CT and σ_{CT} are then uploaded to the Edge Node (EN).

7) $\mathsf{Match}(S_{uid}, (M, \rho)) \rightarrow 1/0$:

This algorithm is executed by the Cloud Storage (CS). A DU submits its attribute set S_{uid}, and the CS checks whether S_{uid} satisfies the access policy (M, ρ). If satisfied, the output is 1; otherwise, the output is 0.

8) $\mathsf{VerifySign}(CT, PK_d, \sigma_{CT}) \rightarrow 1/0$:

This algorithm is executed by the DU. Given CT, PK_d, and σ_{CT}, the DU verifies the correctness of the signature. If the verification succeeds, the output is 1; otherwise, the output is 0.

9) $Decrypt(CT, SK_{uid})) \rightarrow \{m\}$:

This algorithm is executed by the DU. Given the ciphertext set CT and the attribute secret key SK_{uid}, the DU first recovers the symmetric key κ from C_κ, and then uses κ to decrypt C_t, thereby obtaining the original data m.

V. Proposed Scheme

A. Initialization Phase

Input: Security parameter λ

Output: System public parameters GP , master secret key MSK, and public/private key pairs (PK, SK) for each entity

1) Public Parameter Generation

$$G, G_T, g \in G, \quad e : G \times G \rightarrow G_T, \quad MSK = (\alpha, \beta, a, b) \in \mathbb{Z}_p$$

2) Attribute Element Generation

$$h_i = g^{\mathrm{Attr}_i}, \quad GP = (G_T, G, H, g, g^\alpha, e(g, g)^\alpha, h_1, \ldots, h_U)$$

3) Entity Key Generation

Authorization Server AS_i generates its public–private key pair.

$$SK_{aid} = k_{aid} \in \mathbb{Z}_p, \quad PK_{aid} = g^{k_{aid}} \tag{15}$$

Data Owner DO_j generates its public–private key pair.

$$SK_d = k_d \in \mathbb{Z}_p, \quad PK_d = g^{k_d} \tag{16}$$

Each legitimate DU/AS is assigned a unique identifier, denoted as aid and uid.

B. Key Generation Phase

Input: DU identity uid, attribute set $S_{uid} = \{\mathrm{Attr}_1, \ldots, \mathrm{Attr}_U\}$

Output: DU's attribute secret key SK_{uid}

1) Request Phase

DU, characterized by (uid, S_{uid}), submits a key generation request to the primary AS.

The primary AS verifies uid and S_{uid}, obtains a timestamp TS, and computes hash values:

$$t_1 = H(uid \| TS \| 0), t_2 = H(uid \| TS \| 1) \tag{17}$$

The intermediate key is generated as:

$$IC(aid, uid) = \{K_x = h_x^{k_{aid} t_1}, J_x = h_x^{t_2}\}, \quad \forall x \in S_{uid} \tag{18}$$

2) Pre-Preparation Phase

The primary AS forwards the DU's key generation request along with the associated information $\{aid, uid, S_{uid}, IC_{aid,uid}, TS\}$to the backup ASs.

Each backup AS generates attribute key components:

$$K = g^\alpha \cdot g^{a(k_{aid}\beta t_1 + \alpha t_2)} \tag{19}$$

$$L = g^{k_{aid}\beta t_1 + \alpha t_2} \tag{20}$$

$$K'_x = K_x^\beta \cdot g^{b(t_1 + t_2)} \tag{21}$$

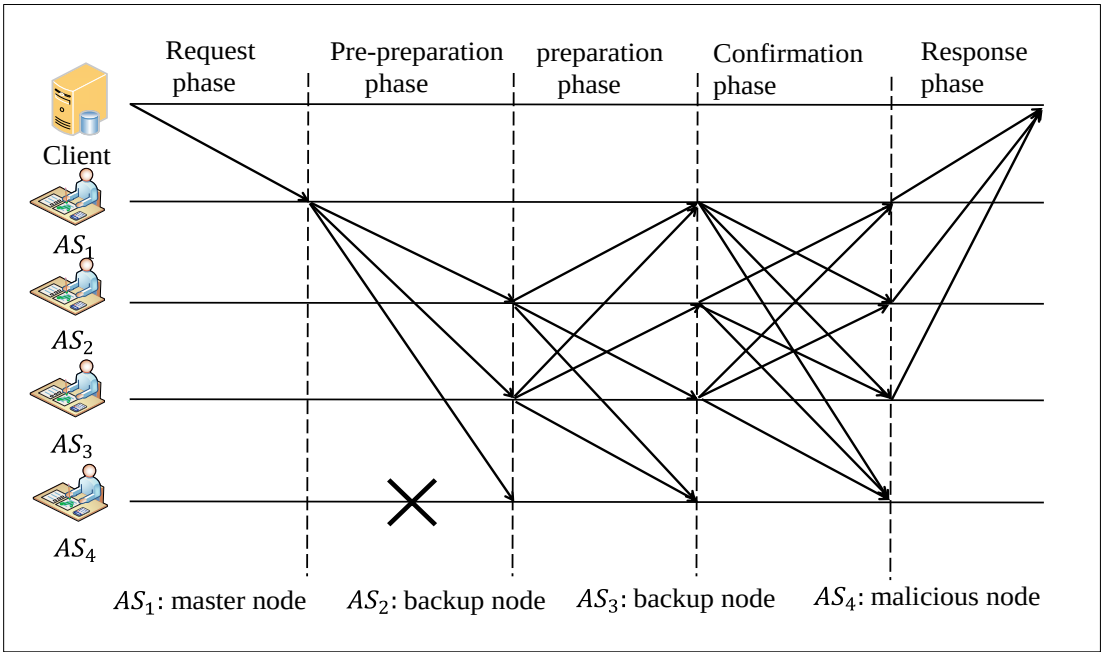


Fig. 2: PBFT Attribute Key Consensus

$$K''_x = J_x^{\alpha} \cdot g^{-b(t_1+t_2)} \tag{22}$$

The complete attribute key is then formed:

$$SK_{uid} = \{K, L, \{K'_x, K''_x\} x \in S_{uid}\}$$

3) Preparation Phase

All ASs verify the attribute key list, compute the new block hash, and broadcast it.

4) Commit Phase

When an AS receives $2f$ consistent values, it broadcasts a commit message.

5) Reply Phase

When DU receives at least $f+1$ consistent replies, the attribute key generation is completed.

The consensus process of PBFT for generating attribute keys is shown in Fig. 2.

C. Encryption Phase

Input:Plaintext m ,access structure (M, ρ)

Output: Ciphertext set $CT = \{C_t, C_\kappa\}$

1) Symmetric encryption of data

DO selects a random symmetric key $\kappa \in G_T$

$$C_t = E_\kappa(m) = \text{Encrypt}(m, \kappa) \tag{23}$$

2) CP-ABE encryption of symmetric key

Choose a secret $s \in \mathbb{Z}_p$ and a random vector

$$\vec{v} = (s, y_2, \ldots, y_n) \in Z_p^n \tag{24}$$

For each row $i = 1, \ldots, l$ of M, compute

$$\lambda_i = M_i \cdot \vec{v}^{\top} \tag{25}$$

Select random $r_i \in Z_p$ for each i

Compute ciphertext components:

$$C = \kappa \cdot e(g,g)^{\alpha s}, C' = g^s \tag{26}$$

$$C_i = (g^a)^{\lambda_i} \cdot h_{\rho(i)}^{-r_i}, D_i = g^{r_i}, \quad \forall i \in \{1, \ldots, l\} \tag{27}$$

3) Output ciphertext set

$$C_\kappa = \{C, C', \{C_i, D_i\}_{i=1}^{l}\}, CT = \{C_t, C_\kappa\} \tag{28}$$

D. Ciphertext Signing and Upload

Input: Ciphertext set $CT = \{C_t, C_\kappa\}$,DO private key $SK_d = k_d$

Output: Signature σ_{CT},ciphertext and metadata uploaded

1) Compute hash of ciphertext set

$$H_{CT} = H(C_t \| C_\kappa) \in Z_p \tag{29}$$

2) Generate BLS signature

$$\sigma_{CT} = g^{H_{CT} \cdot k_d} \tag{30}$$

3) Upload ciphertext and metadata

DO uploads (CT, σ_{CT}) to EN for storage

EN uploads metadata set

$$MD = \{ID, d, Len\} \tag{31}$$

to CS, where ID denotes the storage location of CT on EN, d is the index set, and Len is the ciphertext size.

E. Ciphertext Access and Decryption

Input:Ciphertext set $CT = \{C_t, C_\kappa\}$, signature σ_{CT}, DU attribute set S_{uid},CP-ABE access policy (M, ρ),metadata $MD = \{ID, d, Len\}$

Output:Decrypted message m or access denial

1) Access Permission Verification

DU sends S_{uid} to CS

CS extracts submatrix M_U from M corresponding to S_{uid}

If S_{uid} does not satisfy (M, ρ),output 0 and abort

Otherwise, there exist parameters $\{\omega_i \in Z_p\}_{i \in I}$ such that

$$\mathbf{e} = (1, 0, \ldots, 0) = \sum_{i \in I} \omega_i M_i \tag{32}$$

2) Ciphertext Retrieval

CS queries EN using metadata $MD = \{ID, d, Len\}$

EN returns ciphertext set CT and signature σ_{CT}

3) Signature Verification

Compute hash:

$$H_{CT} = H(C_t \| C_\kappa) \in Z_p \tag{33}$$

Verify BLS signature:

$$e(\sigma_{CT}, g) \stackrel{?}{=} e(PK_d, g)^{H_{CT}} \tag{34}$$

Abort if verification fails; continue if successful

4) Attribute Key Recovery and Symmetric Key Decryption

For each attribute $x \in S_{uid}$ compute:

$$K'_x K''_x = h_x^{k_{aid}\beta t_1 + \alpha t_2} \tag{35}$$

Compute:

$$C_1 = \frac{e(C', K)}{\prod_{i \in I} \left(e(C_i, L) \cdot e(D_i, K'_{\rho(i)} K''_{\rho(i)}) \right)^{\omega_i}} = e(g, g)^{\alpha s} \tag{36}$$

Recover symmetric key:

$$\kappa = \frac{C}{C_1} \tag{37}$$

Decrypt C_t using k to obtain $m = D_\kappa(C_t)$

VI. Security Analysis

In this section, the proposed data sharing scheme is analyzed and formally proven against existing security threats. The results demonstrate that the scheme can effectively meet the security requirements of data sharing in vehicular networks, while maintaining strong security and reliability in the presence of various attacks.

A. Security Analysis

Based on the above security requirements, a theoretical security analysis of the proposed scheme is conducted.

1) Data Confidentiality:

The scheme employs attribute-based encryption (ABE) to enforce access control. Data Owners (DOs) define access policies when encrypting data, ensuring that only Data Users (DUs) whose attributes satisfy the policies can decrypt the ciphertext. In addition, DOs encrypt data using the symmetric AES algorithm to generate a symmetric key κ, which is subsequently encrypted using the CP-ABE scheme. Even if the ciphertext is intercepted by an adversary, the attacker cannot infer DU attributes or recover the plaintext. Unauthorized DUs cannot match the embedded access policies, preventing any plaintext leakage through guessing or attacks. Therefore, the scheme ensures strict confidentiality during data sharing.

Algorithm 1 Multi-Authority CP-ABE with PBFT and BLS Signatures

Require: Security parameter λ, DU uid with attributes S_{uid}, plaintext m, access policy (M, ρ)

Ensure: Ciphertext CT, signature σ_{CT}, decrypted message m (if authorized)

1: **Initialization:** Generate groups G, G_T, generator g, bilinear map e, master key $MSK = (\alpha, \beta, a, b)$

2: Compute attribute elements $h_i = g^{\mathrm{Attr}_i}$ and public parameters $GP = (G_T, G, H, g, g^\alpha, e(g,g)^\alpha, h_1, ..., h_U)$

3: AS_i and DO_j generate key pairs (SK_{aid}, PK_{aid}), (SK_d, PK_d)

4: **Key Generation:**

DU sends key request (uid, S_{uid}); primary AS verifies and computes t_1, t_2

Intermediate keys: $IC = \{K_x = h_x^{k_{aid} t_1}, J_x = h_x^{t_2}\}$, forwarded to backup ASs

Backup ASs compute K, L, K'_x, K''_x and form $SK_{uid} = \{K, L, \{K'_x, K''_x\}\}$ via PBFT consensus

5: **Encryption:**

DO selects symmetric key κ, encrypts $C_t = \mathrm{Encrypt}(m, \kappa)$

CP-ABE: choose s, vector $\vec{v} = (s, y_2, ..., y_n)$, compute $\lambda_i = M_i \vec{v}^\top$, C_i, D_i, $C = \kappa e(g,g)^{\alpha s}$, $C' = g^s$

Ciphertext: $C_\kappa = \{C, C', \{C_i, D_i\}\}$, $CT = \{C_t, C_\kappa\}$

6: **Signing and Upload:**

Compute $H_{CT} = H(C_t \| C_\kappa)$, $\sigma_{CT} = g^{H_{CT} k_d}$

Upload (CT, σ_{CT}) to EN; metadata MD to CS

7: **Decryption:**

DU sends S_{uid} to CS; check $S_{uid} \models (M, \rho)$

Retrieve (CT, σ_{CT}) from EN; verify $e(\sigma_{CT}, g) = e(PK_d, g)^{H_{CT}}$

Recover attribute keys $K'_x K''_x$, compute $C_1 = e(g,g)^{\alpha s}$, recover $\kappa = C/C_1$, decrypt $m = D_\kappa(C_t)$

2) Resistance to AS Collusion Attacks:

Attribute Servers (ASs) employ the PBFT consensus algorithm to distribute key management and generation across multiple nodes. Even if up to f =(n-1)/3 nodes are faulty or malicious, the remaining nodes can achieve correct consensus, ensuring the reliability of key generation. Each AS request is digitally signed, preventing adversaries from forging requests or manipulating the key generation process. Consequently, no single node or small group of malicious ASs can control or reconstruct the complete key, making the scheme resistant to AS collusion attacks.

3) Resistance to DU Collusion Attacks:

The scheme enforces identity binding, where each DU possesses a unique identifier uid, and the attribute key SK_{uid} is tightly bound to this identifier. During decryption, even if multiple DUs attempt to combine their private key components to recover $e(g,g)^{\alpha s}$, the keys generated are unique due to the binding with uid and timestamp TS. A dual verification mechanism ensures that both identity and timestamp are validated during key generation and decryption, preventing colluding DUs from collectively decrypting unauthorized ciphertext.

4) Resistance to EN Forgery Attacks:

DOs randomly select $k_d \in \mathbb{Z}_p$ to generate the signing private key $SK_d = k_d$, which is not included in the ciphertext set CT. Upon receiving CT, DUs obtain the corresponding public key $PK_d = g^{k_d}$ to verify the signature σ_{CT}. Due to the uniqueness of the BLS signature and the randomness of k_d, edge nodes (ENs) cannot forge valid signatures, as computing k_d from the public parameters is computationally infeasible. Thus, the scheme resists EN forgery attacks.

5) Resistance to EN Tampering Attacks:

DUs first encrypt data using AES and then apply BLS signatures to the ciphertext set CT to ensure integrity. During decryption, the DU computes $e(g, g^{k_d})^{H(C_t \| C_\kappa)}$ and verifies the equation

$$e(\sigma_{CT}, g) = e(g, g^{k_d})^{H(C_t \| C_\kappa)} \tag{38}$$

This allows the DU to detect any tampering with CT. Even if a semi-trusted EN modifies the ciphertext, it cannot forge the BLS signature, ensuring the integrity of the data.

B. Security Proof

1) Theorem 1 (Confidentiality Guarantee)

The confidentiality of the proposed scheme is reduced to the decisional q-parallel Bilinear Diffie-Hellman Exponent (q-parallel BDHE) assumption in bilinear groups (formally defined in Section III).

The decisional q-parallel BDHE problem is stated as follows: Given a group element vector $\vec{y} = \left(g, g^a, g^{a^2}, \ldots, g^{a^q}, g^b, \{g^{a^i b^j}\}_{i,j \le q, j \ne i}\right) \in G^{O(q^2)}$ and an element $T \in G_T$, decide whether $T = e(g,g)^{a^{q+1}b}$ or T is a random element in G_T. The q-parallel BDHE assumption states that no probabilistic polynomial-time (PPT) adversary can distinguish these two cases with non-negligible advantage.

If there exists a PPT adversary $\mathcal{A}$ that breaks the IND-CPA security of the proposed scheme with non-negligible advantage ε, then we construct a simulator $\mathcal{B}$ that solves the decisional q-parallel BDHE problem with the same advantage.

Proof via a hybrid argument:

- **Game**$_0$: The real IND-CPA security experiment where $\mathcal{A}$ interacts with the actual scheme.
- **Game**$_1$ **(Hardness instance embedding)**: $\mathcal{B}$ is given a q-parallel BDHE instance $(\vec{y}, T)$. $\mathcal{B}$ simulates the public parameters by embedding the challenge as follows:
  - Implicitly set the master secret $\alpha = a^{q+1}$ (without explicitly computing it);
  - Program $e(g,g)^\alpha = e(g,g)^{a^{q+1}}$ and attribute elements h_i using terms from $\vec{y}$;
  - All public parameters are distributed identically to the real scheme.

  Thus, $\mathcal{A}$ cannot distinguish Game$_0$ from Game$_1$.
- **Game**$_2$ **(Challenge ciphertext construction)**: When $\mathcal{A}$ submits two equal-length messages m_0, m_1, $\mathcal{B}$ chooses $b \xleftarrow{R} \{0,1\}$, sets the secret share $s = b$ (taken from the challenge instance), and constructs the challenge ciphertext as

$$C = m_b \cdot T \cdot e(g^{\alpha'}, g^s), \qquad C' = g^s. \tag{39}$$

If $T = e(g,g)^{a^{q+1}b}$, then $C = m_b \cdot e(g,g)^{\alpha s}$, which is a valid encryption of m_b. If T is random, C hides m_b perfectly in $\mathcal{A}$'s view.

All private key queries issued by $\mathcal{A}$ are answered consistently using polynomial simulation over $\vec{y}$, without generating any term involving a^{q+1}.

Therefore, the advantage of $\mathcal{A}$ in distinguishing Game$_1$ from Game$_2$ is exactly the advantage of $\mathcal{B}$ in deciding whether $T = e(g,g)^{a^{q+1}b}$ or not.

This completes the reduction. Any non-negligible advantage ε of $\mathcal{A}$ in breaking the confidentiality of the scheme implies a non-negligible advantage ε for $\mathcal{B}$ in solving the q-parallel BDHE problem, which contradicts the hardness assumption.

Thus, the proposed scheme is IND-CPA secure under the decisional q-parallel BDHE assumption in the standard model.

2) Theorem 2 (Collusion Resistance)

In PBFT consensus, colluding AS nodes cannot cause a fork.

In a PBFT network with n nodes, up to $f = (n-1)/3$ nodes may be faulty or malicious.

If a fork occurs, the malicious set I must form majorities simultaneously with two disjoint honest sets P_1, P_2:

$$|P_1| + |I| \ge n - f, |P_2| + |I| \ge n - f \tag{40}$$

In the extreme case $|I| = f$, we have $|P_1|, |P_2| \ge n - 2f$, which implies

$$|P_1| + |P_2| \ge 2n - 4f \tag{41}$$

Given the total number of nodes $|P_1| + |P_2| + |I| = n$ and $|I| = f$ a contradiction arises:

$$n \le 3f \tag{42}$$

However, according to PBFT,$f = (n-1)/3$, which makes this inequality impossible.

Thus, even if the malicious AS nodes collude, they cannot control both majorities, and a fork cannot occur.

3) Security Conclusion 3 (Key Misuse Resistance)

During key generation, colluding AS nodes may attempt to abuse keys. If $AS_{s_1}$and AS_{s_2}generate intermediate keys for the same DU:

$$K_1 = g^\alpha \cdot g^{a\,(k_{\mathrm{aid}_1}\beta t_1 + \alpha t_2)}, K_2 \quad = g^\alpha \cdot g^{a\,(k_{\mathrm{aid}_2}\beta t_1 + \alpha t_2)}. \tag{43}$$

$$L_1 = g^{k_{\mathrm{aid}_1}\beta t_1 + \alpha t_2}, L_2 = g^{k_{\mathrm{aid}_2}\beta t_1 + \alpha t_2} \tag{44}$$

$$KK_{x1} = h_x^{k_{\mathrm{aid}_1}\beta t_1 + \alpha t_2}, KK_{x2} = h_x^{k_{\mathrm{aid}_2}\beta t_1 + \alpha t_2} \tag{45}$$

The combined ratio yields:

$$K_1/K_2 = g^{(k_{\mathrm{aid}_1} - k_{\mathrm{aid}_2})\beta t_1} \tag{46}$$

$$L_1/L_2 = g^{(k_{\mathrm{aid}_1} - k_{\mathrm{aid}_2})\beta t_1} \tag{47}$$

$$KK_{x_1}/KK_{x_2} = h_x^{(k_{\mathrm{aid}_1} - k_{\mathrm{aid}_2})\beta t} \tag{48}$$

Attempt to compute g^α:

$$g^\alpha = \left(L_1/(L_1/L_2)^{k_{\mathrm{aid}_1}/(k_{\mathrm{aid}_1} - k_{\mathrm{aid}_2})}\right)^{1/t_2} \tag{49}$$

Uniqueness of timestamps prevents valid combination, hence colluding AS nodes cannot derive usable intermediate keys. Scheme resists key misuse attacks.

VII. Performance Evaluation

This section presents the experimental work conducted to demonstrate the effectiveness of the proposed method. First, the proposed scheme is compared with several typical ABE schemes in terms of functionality, storage overhead, and computational overhead. Next, the average time overhead of each phase in the proposed scheme is measured, and the experimental results are analyzed. Finally, the transaction delay and communication overhead on the key blockchain are simulated and analyzed.

A. Performance Analysis

1) Functional Comparison Analysis:

Based on the functionalities implemented in the proposed scheme, this section compares the proposed scheme with recent schemes [23], [24], [25], [18], [27], [29], [30]. Table II presents the results of this comparison. The results indicate that schemes [24], [27], [29] do not support cloud-edge collaboration. Although schemes [23], [25], [18], [30]support cloud-edge collaboration, the ciphertext is still stored in the cloud server (CS), which does not fundamentally address the single-point-of-failure problem in data storage. In the proposed scheme, the ciphertext is stored on edge nodes (ENs) while metadata is stored in the CS. Data users (DUs) can download ciphertext from nearby ENs, which not only resolves the single-point-of-failure issue but also meets the DUs' low-latency requirements.

While schemes [18], [29]and the proposed scheme support multiple authorization servers, schemes [18], [29] do not integrate multiple authorization servers with blockchain, making them unable to resist key abuse. Moreover, only schemes [25], [30] and the proposed scheme employ signatures to ensure that data remains unaltered during the sharing process, thereby guaranteeing data integrity. In addition, except for scheme [25], all other schemes support fine-grained access control. Although both scheme [25] and the proposed scheme utilize blockchain, their roles differ: the proposed scheme leverages PBFT in combination with blockchain to mitigate key abuse and collusion attacks among multiple authorization servers, whereas scheme [25]merely records and verifies the data-sharing process to prevent malicious tampering.

In summary, compared with existing schemes, the proposed scheme implements more practical functionalities, making it better suited to the real-world data-sharing requirements in vehicular networks.

2) Computational Overhead Analysis:

This work focuses on the computational overhead of the algorithms in the key generation, encryption, and decryption phases. In this section, the proposed scheme is compared with schemes [23], [18], [29]. Here, $|I_s||$ denotes the number of attributes involved in the decryption phase, and $|S|$ represents the number of attributes in the attribute set s.Pand E denote bilinear mapping and exponentiation operations, respectively.

As shown in Table III, the computational overhead in the key generation phase for schemes [23], [18], [29] and the proposed scheme is proportional to the number of attributes. However, the proposed scheme has the smallest coefficient, resulting in the lowest overhead during key generation because each authorization server (AS) only needs to generate its own key pair without generating key pairs for other entities.

In the encryption and decryption phases, schemes [23], [18], [29] involve more bilinear pairings and exponentiation operations, leading to relatively higher computational overhead. In contrast, the proposed scheme processes ciphertexts and key ciphertexts separately, reducing the complexity of data processing, and requires fewer bilinear mapping and exponentiation operations than the other schemes. Consequently, it exhibits the lowest computational overhead.

In summary, the proposed scheme achieves minimal computational overhead across the key generation, encryption, and decryption phases. Therefore, compared with the aforementioned schemes, it demonstrates a significant advantage in computational performance.

3) Resistance to DU Collusion Attacks:

Different cloud-edge storage schemes exhibit significant differences in system resource consumption. In this section, we compare the proposed scheme with Schemes [23], [18], [29] in terms of storage requirements, focusing on the space consumed by system initialization parameters, attribute keys, and ciphertexts. The considered factors include the lengths of group elements $|G|, |G_T|$, and $|Z_p|$, the size of the symmetric encryption ciphertext $|M_{SE}|$, the number of attributes required for decryption $|I_S|$, the depth of the access tree T, and the number of rows l in the access policy matrix M .

As shown in Table IV, the proposed scheme achieves the lowest storage overhead for both initialization parameters and ciphertexts. This advantage is primarily due to the distributed management of attribute keys across multiple authorities, which avoids the redundancy caused by centralized storage. Moreover, by employing CP-ABE to encrypt only the symmetric key rather than the entire data, the ciphertext size depends on the complexity of the access policy rather than the size of the original data.

Regarding key storage, the proposed scheme exhibits similar overhead to Scheme [29]. This is because both schemes adopt a similar key structure, where each entity stores only the portion of the key relevant to its attributes. Therefore, the storage cost is related to the user's attribute set $|I_S|$. Overall, through distributed key management and optimized encryption strategies, the proposed scheme significantly reduces the overall storage cost, providing an efficient solution for large-scale vehicular network data management.

TABLE II: Functional Comparison

Function	Scheme [29]	Scheme [18]	Scheme [23]	Scheme [27]	Scheme [25]	Scheme [24]	Scheme [30]	ours
Cloud-Edge Collaboration	✗	✓	✓	✗	✓	✗	✓	✓
Multiple Authorization Authorities	✓	✓	✗	✗	✗	✗	✗	✓
BLS Signature	✗	✗	✗	✗	✓	✗	✓	✓
Access Structure	✓	✓	✓	✓	✗	✗	✓	✓
Fine-Grained Access Control	✓	✓	✓	✓	✗	✓	✓	✓
Blockchain	✗	✗	✗	✗	✓	✗	✗	✓
Key Misuse Resistance	✗	✗	✓	✗	✗	✗	✗	✓
Collusion Attack Resistance	✗	✓	✓	✓	✗	✓	✓	✓

TABLE III: Computational Overhead Comparison

Scheme	Initialization Parameters	Key	Ciphertext
Scheme [29]	$(5\|S\|+4)E$	$(5\|I_S\|+1)E+P$	$3\|I_S\|E+(3\|I_S\|+1)P$
Scheme[18]	$(2\|S\|+4)E$	$(4\|I_S\|+1)E+3\|I_S\|P$	$(\|I_S\|+2)E+(2\|I_S\|+1)P$
Scheme[23]	$(3\|S\|+4)E$	$(2\|I_S\|+4)E+\|I_S\|P$	$(2\|I_S\|+1)E+(2\|I_S\|)P$
Ours	$(2\|S\|+4)E$	$(2\|I_S\|+1)E+P$	$(\|I_S\|+2)E+(\|I_S\|+1)P$

TABLE IV: Storage Overhead Comparison

Scheme	Initialization Parameters	Key	Ciphertext
Scheme [29]	$5\|G\|+\|G_T\|$	$2(\|I_S\|+1)\|G\|$	$(3l+1)\|G\|+\|G_T\|$
Scheme [18]	$3\|G\|+\|G_T\|$	$(\|I_S\|+1)\|Z_p\|$	$3\|G\|+2\|G_T\|+\|M_{SE}\|+2\|Z_p\|$
Scheme [23]	$3\|G\|+\|G_T\|$	$(2\|I_S\|+5)\|G\|+\|Z_p\|$	$\|T\|+4\|G\|+3\|G_T\|$
Ours	$2\|G\|+\|G_T\|$	$2(\|I_S\|+1)\|G\|$	$(l+1)\|G\|+\|M_{SE}\|+\|G_T\|$

TABLE V: Average Execution Time of Major Cryptographic Operations

Operation	Time (ms)
Bilinear Pairing P	4.190024
Exponentiation Operation E	0.743038

B. Experimental Results Analysis

To evaluate the efficiency of the proposed scheme in vehicular network scenarios, this study assesses the response times of critical cryptographic operations, including key generation, data encryption, and decryption. Beyond measuring the duration of each stage, the performance of different schemes under the same tasks was compared to quantify the practical advantages of the proposed approach. In addition, data transmission times under different storage strategies were independently tested, comparing the proposed scheme with Schemes [23], [29].

The experiments were conducted on a Linux Ubuntu 64-bit platform using the Charm-Crypto framework, with the elliptic curve "MNT224". The hardware setup included an AMD Ryzen 7 4800H processor (2.90 GHz, with Radeon Graphics) and 8 GB of RAM. To reduce random errors, each experiment was repeated 500 times, and the average execution time of each key cryptographic operation was recorded. The results are presented in tabular form (see Table V), providing insights into the scheme's operational efficiency in real-world deployment.

Compared to conventional schemes, the proposed approach exhibits shorter and more stable response times across all critical operations, demonstrating its suitability and advantages in high-frequency request environments.

1) Time Overhead in the Key Generation Phase:

As shown in Fig. 3, scheme [29] exhibits the fastest increase in time overhead during the key generation phase; when the number of attributes reaches 25, the time overhead approaches 100 ms. This indicates that the scheme has high computational complexity and relatively low efficiency when handling a large number of attributes. Scheme [23] optimizes the mapping between attributes and key generation, reducing unnecessary computation. At 25 attributes, its time overhead is approximately 60 ms. Although it also increases with the number of attributes, its growth rate is slower than that of scheme [29] and slightly faster than that of scheme [18] and the proposed scheme.

The time overhead of scheme [18] nearly coincides with that of the proposed scheme, but it is still slightly higher. In the proposed scheme, the time overhead increases more gradually during key generation; with 25 attributes, the overhead is about 40 ms. Even when handling a larger number of attributes, it maintains relatively low time overhead. This efficiency is achieved because multiple authorization servers (ASs) perform the process independently, distributing the computational load, and each AS only needs to generate its own key pair.

Therefore, the proposed scheme demonstrates superior computational performance in the key generation phase, with average time overhead reductions of 58.62%, 2.30%, and 32.80% compared to schemes [29], [18], and [23], respectively.

2) Time Overhead in the Encryption Phase:

As shown in Fig. 4, during the encryption process, all schemes exhibit an increasing trend in time overhead as the number of attributes grows, due to the higher computational load and complexity. Among them, scheme [18] shows the fastest growth and consistently incurs higher time overhead than the other schemes. When the attribute count reaches 25,

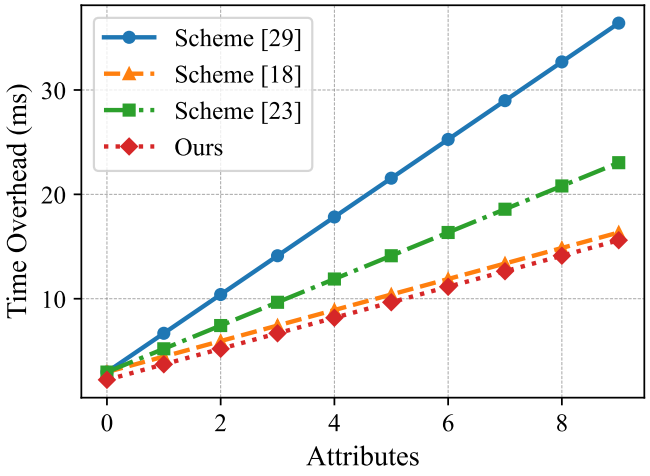


Fig. 3: Key Generation Time

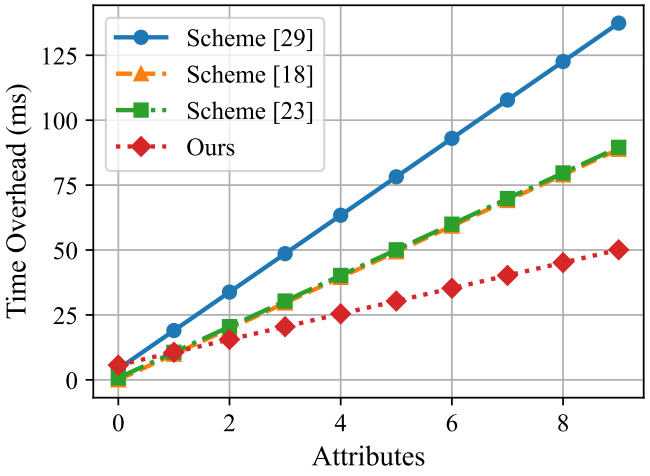


Fig. 5: Decryption Time

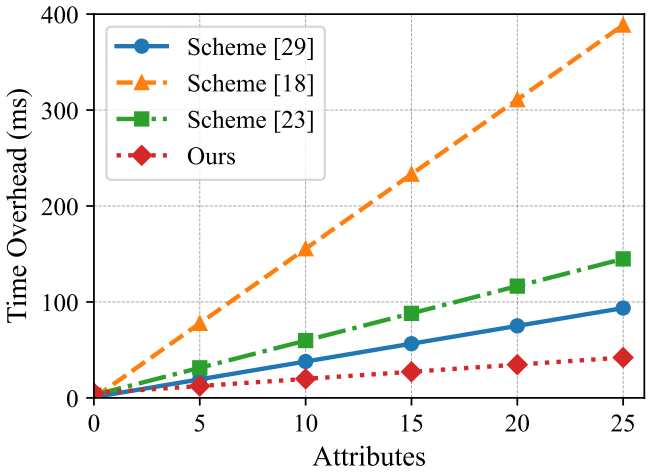


Fig. 4: Encryption Time

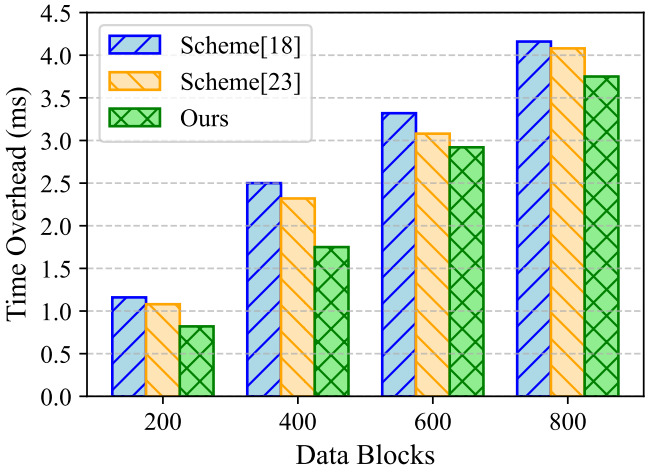


Fig. 6: Edge Node Transmission Time

its time overhead approaches 350 ms, the highest value in the figure. This is because scheme [18] involves numerous parameter and attribute computations in the encryption phase, and the computational cost is proportional to the number of attributes, resulting in relatively high overall time overhead.

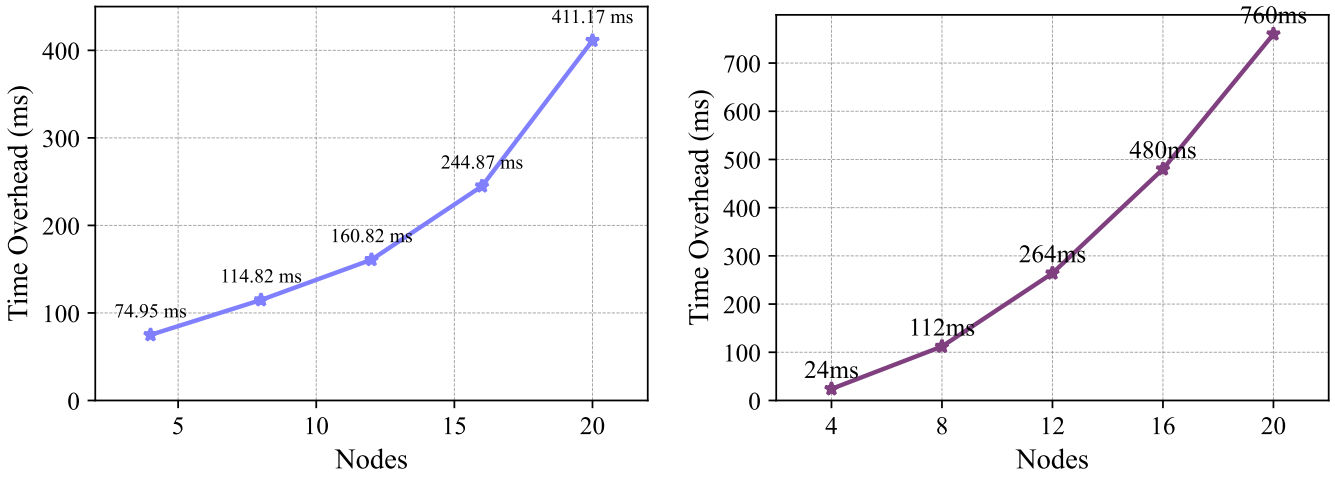
In contrast, the proposed scheme maintains low computational overhead even with a large number of attributes. This is achieved because, in the encryption process, the proposed scheme only uses the CP-ABE algorithm to encrypt the symmetric key, significantly reducing the number of bilinear mapping and exponentiation operations. The average time overhead of the proposed scheme in the encryption phase is reduced by 47.35%, 92.15%, and 66.57% compared with schemes [29], [18], and [23], respectively. These results demonstrate that the proposed scheme has a clear advantage in the encryption phase.

3) Time Overhead in the Decryption Phase:

As shown in Fig. 5, during the decryption phase, the time overhead of all schemes increases with the number of attributes. Scheme [29] exhibits the fastest growth because the decryption process involves multiple steps of computation and verification. In particular, handling complex access policies and key structures requires substantial computation to ensure the correctness and security of decryption.

Schemes [18] and [23] show similar time overhead for certain attribute counts, as they perform the same number of bilinear pairing operations. In contrast, the proposed scheme exhibits the slowest growth in decryption time overhead. This is because it can quickly verify attribute satisfaction and only requires a single bilinear pairing operation, avoiding the multiple nested computations typical of traditional CP-ABE schemes.

In terms of decryption performance, the average time overhead of the proposed scheme is reduced by 65.19%, 43.18%, and 47.21% compared with schemes [29], [18], and [23], respectively. The results in Fig. 5clearly demonstrate that the proposed scheme has a significant advantage in the decryption phase.



(a) (b)

Fig. 7: Key Blockchain Performance
(a) Transaction Delay; (b) Communication Overhead

4) Transmission time

To quantify the transmission performance of different cloud-edge collaborative storage schemes, this study standardizes the link speeds as follows: $DO \rightarrow EN$ at 1 Mb/s, and EN→CS/DU at 512 Kb/s. The transmission procedures for the three schemes are as follows:

Scheme [18]: Ciphertext is uploaded to the CS, which generates metadata and then sends it to the EN;

Scheme [23]: Ciphertext is uploaded to the CS, from which the EN downloads the ciphertext and forwards it to the DU;

Ours: Ciphertext and signatures are uploaded directly to the EN, which transmits only metadata to the CS; when the DU accesses the ciphertext, it can retrieve it directly from the nearest EN.

Scheme [29] is not included in this transmission time comparison due to its lack of edge nodes and reliance solely on centralized cloud storage.

The experimental results indicate that our scheme achieves the shortest data transmission time (Fig. 6), reducing it by 22.40% compared to Scheme [18] and by 17.35% compared to [23]. Therefore, our approach demonstrates clear advantages in latency-sensitive vehicular network environments.

5) Blockchain Consensus Performance Evaluation

To rigorously evaluate the blockchain component for attribute key consensus, we implemented a complete PBFT-based permissioned blockchain prototype using the Rust programming language on an Intel® Core™ i7-10700 platform with 16 GB RAM. Unlike public blockchains (e.g., Ethereum), where Gas consumption is a primary economic metric, our permissioned design incurs no monetary costs. Therefore, transaction latency (consensus delay) and communication overhead are selected as the primary performance indicators for practical feasibility in IoV key management. Realistic workloads were simulated by issuing 500 client requests across varying numbers of Authorization Servers (ASs).

a) Transaction Delay: Transaction delay refers to the time interval between a data user (DU) sending a transaction request to the primary node and the DU confirming consensus. Multiple tests were conducted, averaging 100 transaction delays for reliability. As shown in Fig. 7a, transaction latency increases with the number of ASs, consistent with PBFT's quadratic communication complexity. However, even at 20 ASs—a realistic upper bound for regional traffic authorities in practical IoV deployments—the latency remains acceptable for infrequent attribute key management operations.

b) Communication Overhead: Communication overhead refers to the network traffic generated by the ASs during the consensus process. Assuming there are n ASs in the system, the communication overhead for each stage is analyzed as follows:

Pre-prepare stage: The primary AS sends pre-prepare messages to all other ASs, resulting in a communication overhead of $n-1$.

Prepare stage: Each AS sends prepare messages to all other ASs, incurring a communication overhead of $(n-1)(n-1)$.

Commit stage: Each AS generates commit messages and broadcasts them to the other nodes, leading in a communication overhead of $n(n-1)$.

Consequently, the total communication overhead for the ASs to complete the PBFT consensus algorithm is:

$$N = (n-1) + (n-1)(n-1) + n(n-1) = 2n(n-1). \quad (50)$$

Due to the permissioned nature of the blockchain, no Gas consumption is incurred. The measured communication overhead, although quadratic, remains reasonable and manageable for the limited scale of ASs (typically 10–20 per region) in real-world IoV systems, as deployed on resource-rich regional servers rather than constrained edge devices.

VIII. Conclusion

This paper presents a cloud-edge collaborative secure data sharing framework for the Internet of Vehicles (IoV). By offloading cloud storage tasks to edge nodes, the scheme effectively mitigates data transmission latency and single-point failure issues while meeting low-latency access requirements. To ensure data security, CP-ABE is employed for attribute-based access control, and a multi-authority architecture combined with blockchain and PBFT consensus optimizes attribute key generation and distribution, enhancing resistance to key misuse and collusion attacks. BLS short signatures are further applied to verify ciphertext integrity, ensuring data remains tamper-proof during transmission. Comprehensive security analysis and experimental evaluation demonstrate that the proposed scheme significantly improves data confidentiality, integrity, and overall system performance while keeping the overhead of blockchain-based key consensus manageable and isolated from resource-constrained edge nodes.

References

[1] S. R. Gudimetla, "Data encryption in cloud storage," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 6, pp. 2582–5208, 2024.

[2] F. Shang and X. Deng, "A data sharing scheme based on blockchain for privacy protection certification of Internet of Vehicles," *Vehicular Communications*, vol. 51, Art. no. 100864, 2025.

[3] Y. Yang, R. Shi, K. Li, *et al.*, "Multiple access control scheme for EHRs combining edge computing with smart contracts," *Future Generation Computer Systems*, vol. 129, pp. 453–463, 2022.

[4] J. Qin, Y. Xun, and J. Liu, "Cvmids: Cloud-vehicle collaborative intrusion detection system for Internet-of-Vehicles," *IEEE Internet of Things Journal*, vol. 11, no. 1, pp. 321–332, 2023.

[5] X. Shi, Y. Guo, W. Jin, *et al.*, "Toward forward-secure end-to-end data sharing: an attribute-key-free CP-ABE scheme," in *Proc. ICASSP 2025 IEEE Int. Conf. on Acoustics*, 2025, pp. 1–5.

[6] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Security and Privacy (SP'07)*, 2007, pp. 321–334.

[7] V. Goyal, O. Pandey, A. Sahai, *et al.*, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Computer and Communications Security*, 2006, pp. 89–98.

[8] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proc. 14th ACM Conf. Computer and Communications Security*, 2007, pp. 456–465.

[9] A. Lewko, T. Okamoto, A. Sahai, *et al.*, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Advances in Cryptology–EUROCRYPT 2010*, Springer, 2010, pp. 62–91.

[10] S. Wang, K. Liang, J. K. Liu, *et al.*, "Attribute-based data sharing scheme revisited in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1661–1673, 2016.

[11] X. Liu, W. Chen, and Y. Xia, "Security-aware information dissemination with fine-grained access control in cooperative multi-RSU of VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2170–2179, 2020.

[12] N. Chen, J. Li, Y. Zhang, *et al.*, "Efficient CP-ABE scheme with shared decryption in cloud storage," *IEEE Transactions on Computers*, vol. 71, no. 1, pp. 175–184, 2020.

[13] Z. Zhang and X. Ren, "Data security sharing method based on CP-ABE and blockchain," *Journal of Intelligent & Fuzzy Systems*, vol. 40, no. 2, pp. 2193–2203, 2021.

[14] S. Das and S. Namasudra, "Multiauthority CP-ABE-based access control model for IoT-enabled healthcare infrastructure," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 821–829, 2022.

[15] Z. Guo, G. Wang, Y. Li, *et al.*, "Accountable attribute-based data-sharing scheme based on blockchain for vehicular ad hoc network," *IEEE Internet of Things Journal*, vol. 10, no. 8, pp. 7011–7026, 2022.

[16] L. Zhang, Y. Zhang, Q. Wu, *et al.*, "A secure and efficient decentralized access control scheme based on blockchain for vehicular social networks," *IEEE Internet of Things Journal*, vol. 9, no. 18, pp. 17938–17952, 2022.

[17] S. J. Horng, C. C. Lu, and W. Zhou, "An identity-based and revocable data-sharing scheme in VANETs," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15933–15946, 2020.

[18] J. Liu, Y. Li, R. Sun, *et al.*, "SDSS: Secure data sharing scheme for edge enabled IoV networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 11, pp. 12038–12049, 2023.

[19] B. Jiang, C. Li, Y. Tang, *et al.*, "Secure cross-chain transactions for medical data sharing in blockchain-based Internet of Medical Things," *International Journal of Network Management*, vol. 35, no. 1, Art. no. e2279, 2025.

[20] J. Li, D. Han, T. H. Weng, *et al.*, "A secure data storage and sharing scheme for port supply chain based on blockchain and dynamic searchable encryption," *Computer Standards & Interfaces*, vol. 91, Art. no. 103887, 2025.

[21] L. Zhang, Z. Ou, C. Hu, *et al.*, "Data sharing in the metaverse with key abuse resistance based on decentralized CP-ABE," *IEEE Transactions on Computers*, 2024.

[22] S. Li, K. Niu, and B. Wu, "A blockchain-based secure data sharing scheme with efficient attribute revocation," *Journal of Systems Architecture*, vol. 159, Art. no. 103309, 2025.

[23] T. Liu, Z. Li, Y. Ji, *et al.*, "Efficient key-escrow-free and vehicle-revocable data sharing protocol for vehicular ad hoc network," *IEEE Internet of Things Journal*, vol. 11, no. 7, pp. 11540–11553, 2023.

[24] H. Deng, Z. Qin, Q. Wu, *et al.*, "Achieving fine-grained data sharing for hierarchical organizations in clouds," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 2, pp. 1364–1377, 2022.

[25] Z. Wang, J. Wang, Y. Liu, *et al.*, "Privacy-preserving attribute-based access control scheme with intrusion detection and policy hiding for data sharing in VANET," *IEEE Internet of Things Journal*, vol. 11, no. 13, pp. 23348–23369, 2024.

[26] X. Zhou, D. He, J. Ning, *et al.*, "AADEC: Anonymous and auditable distributed access control for edge computing services," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 290–303, 2022.

[27] K. Yang, J. Shu, and R. Xie, "Efficient and provably secure data selective sharing and acquisition in cloud-based systems," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 71–84, 2022.

[28] B. Xie, Y. P. Zhou, X. Y. Yi, *et al.*, "An improved multi-authority attribute access control scheme based on blockchain and elliptic curve for efficient and secure data sharing," *Electronics*, vol. 12, no. 7, Art. no. 1691, 2023.

[29] C. Zhao, L. Xu, J. Li, *et al.*, "Toward secure and privacy-preserving cloud data sharing: Online/offline multiauthority CP-ABE with hidden policy," *IEEE Systems Journal*, vol. 16, no. 3, pp. 4804–4815, 2022.

[30] F. Wang, J. Cui, Q. Zhang, *et al.*, "Blockchain-based secure cross-domain data sharing for edge-assisted industrial Internet of Things," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 3892–3905, 2024.

[31] J. Li, Y. Fan, X. Bian, *et al.*, "Online/offline MA-CP-ABE with cryptographic reverse firewalls for IoT," *Entropy*, vol. 25, no. 4, Art. no. 616, 2023.

[32] S. Roy, J. Agrawal, A. Kumar, *et al.*, "Mh-ABE: Multi-authority and hierarchical attribute based encryption scheme for secure electronic health record sharing," *Cluster Computing*, vol. 27, no. 5, pp. 6013–6038, 2024.

[33] P. Duan, Z. Ma, H. Gao, *et al.*, "Multi-authority attribute-based encryption scheme with access delegation for cross blockchain data sharing," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 323–337, 2025.

[34] W. Li, C. Feng, L. Zhang, *et al.*, "A scalable multi-layer PBFT consensus for blockchain," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1146–1160, 2020.

[35] Z. Li, J. Wang, and Y. Li, "An improved PBFT consensus algorithm based on reputation and gaming," *The Journal of Supercomputing*, vol. 81, no. 1, pp. 323, 2025.