Secure and Efficient Personalized Multi-Receiver Data Sharing with Cross-Domain Authentication for Internet of Vehicles

Taolong Su, Guanjie Cheng, Junqin Huang, Xinkui Zhao, and Shuiguang Deng, *Senior Member, IEEE*

***Abstract*—The Internet of Vehicles (IoV) has the potential to greatly improve traffic efficiency and road safety through the real-time exchange of data between vehicles and transport infrastructures. While many schemes employ Certificateless Signcryption (CLSC) to ensure data confidentiality and provide authentication in IoV, existing methods are predominantly designed for single-sender/single-receiver or multi-sender/single-receiver configurations. There is a notable gap in the research addressing the personalized sharing of data with multiple specific users. This paper addresses the practical and challenging problem of personalized multi-receiver data sharing, where customized data is sent to each recipient while remaining confidential from others. Furthermore, the dynamic and heterogeneous nature of vehicles introduces significant challenges for authentication across diverse geographic domains. To address these issues, we propose a novel, lightweight personalized multi-receiver CLSC algorithm featuring cross-domain authentication, thereby enhancing its applicability across various IoV scenarios. Our scheme also reduces computational complexity by enabling aggregated unsigncryption. Additionally, we introduce a pseudonym generation mechanism to ensure the anonymity of participants' identities while maintaining traceability. Rigorous formal security analyses demonstrate that our scheme satisfies all specified security requirements. Furthermore, the experimental evaluations confirm the efficiency and practicality of our scheme.**

***Index Terms*—IoV, multi-receiver data sharing, certificateless signcryption, cross-domain authentication, traceability.**

I. Introduction

THE Internet of Vehicles (IoV) has recently gained significant attention due to the rapid advancements in Intelligent Transportation System (ITS) [1], [2]. IoV facilitates connectivity among smart vehicles, Roadside Units (RSUs), and other ITS infrastructure, forming an integrated network that shares traffic data efficiently to enhance road safety and optimize driving experience [3]. For instance, in the event of a traffic accident, involved vehicles and passersby can issue warnings to nearby vehicles, enabling them to avoid the site and make way for emergency services. Recognizing its potential, nations worldwide are actively promoting IoV deployment [4].

While IoV promises a future of efficient and safe traffic, it also poses significant security challenges. The openness and heterogeneity of wireless communication mediums makes IoV susceptible to adversaries who can easily infiltrate the network to conduct malicious activities such as eavesdropping, forging, and data manipulation [2]. These actions can have dire consequences. For instance, an adversary broadcasting a fake traffic accident can mislead drivers, causing unnecessary diversions and potential traffic jams. Moreover, these vulnerabilities seriously compromise drivers' privacy, such as real-time location, travel history, and personal habits, leading to unpredictable security threats. To address these challenges, robust security mechanisms for data sharing in IoV are imperative.

The hybrid encryption model, which encrypts the exchanged data using symmetric encryption followed by public-key encryption to secure the symmetric key, has traditionally been the industry standard for addressing security concerns in IoV [5]–[7]. However, these methods can heavily drain the resources of IoT devices and often fail to meet the low-latency requirements of IoV applications. To address these issues, signcryption—a cryptographic primitive that integrates encryption and signature functions into a single logical step, has been introduced, significantly reducing computational overhead [8]. To further eliminate certificate management and key escrow issues, Certificateless Signcryption (CLSC) has been proposed, offering a more efficient solution for ensuring confidentiality and lightweight authentication in IoV [9].

A. Motivation

Traditional data sharing strategies in IoV typically focus on either single-sender/single-receiver (one-to-one) [10], [11] or multi-sender/single-receiver (many-to-one) configurations [12]–[14]. The former mode is often employed for delivering specific messages to designated destination, whereas the latter involves multiple senders communicating with a single receiver. In addition, there are one-to-many data sharing scenarios, such as broadcast and multicast, where the same data is sent to a group of recipients simultaneously. However, these existing one-to-many schemes fail to provide personalized data sharing, where each receiver gets a customized message. For example, in a busy urban intersection, different vehicles require distinct information: Vehicle A needs details about the traffic accident that happened 1 km ahead to choose a detour route, Vehicle B needs information on road construction situation 500 m ahead and adjust its driving speed, and Vehicle C needs the location of the nearest parking space. Instead of broadcasting generic traffic information to

Taolong Su, Guanjie Cheng, Xinkui Zhao, and Shuiguang Deng are with the School of Computer Science and Technology, Zhejiang University, Hangzhou 310007, China (e-mail: cyning7357@126.com; chengguanjie@zju.edu.cn; zhaoxinkui@zju.edu.cn; dengsg@zju.edu.cn).

Junqin Huang is with the School of Cyber Science and Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China (e-mail: junqin.huang@sjtu.edu.cn).

(Corresponding authors: Guanjie Cheng, Shuiguang Deng.)

all nearby vehicles, the RSU should customize messages for each vehicle to enhance traffic efficiency and service quality. One-to-one communication in IoV is too time-consuming and may lead to bandwidth congestion. Personalized multi-receiver data sharing allows users to send customized messages to multiple specified receivers, offering more flexibility than the traditional one-to-many mode and conserving more resources than the one-to-one mode [15]. The exchanged data must remain confidential to everyone except the specific receiver. Additionally, the real-time requirement and resource-limited nature of IoV are supposed to be satisfied [16]. To tackle these challenges, we propose a secure and efficient personalized multi-receiver data sharing scheme by designing a novel CLSC algorithm.

Vehicles in IoV frequently travel across multiple Geographic Domains (GDs), such as cities, states, or even countries. Each GD may have its own set of security policies, administrative controls, and trust infrastructures. Ensuring seamless and secure authentication as vehicles cross these boundaries is crucial for maintaining continuous IoV services. Moreover, the real-time nature of IoV demands immediate and reliable cross-domain authentication. Delayed or ineffective authentication can result in the inability to prevent malicious activities, such as unauthorized access. Most cross-domain authentication solutions are based on either Public Key Infrastructure (PKI) with Certificate Authority (CA) [17], [18] or Identity-Based Cryptography (IBC) [19], [20]. However, PKI imposes substantial overhead due to the cumbersome management of digital certificates, while IBC incurs significant computational overhead primarily due to its reliance on bilinear pairings and is criticized for key escrow issues [21]. In comparison, CLSC is a superior choice as it eliminates the need for certificate management and resolves key escrow problems.

Additionally, data shared during a journey in IoV such as speed, direction and location, is inherently sensitive and critical to an individual's privacy. Malicious attackers might track these data and exploit it to harm the drivers [6]. Pseudonymization is an effective method to address this issue by concealing the real identities of data owners. However, blindly covering up users' identities can also make it difficult to trace malicious users [15], [22]. Therefore, we introduce pseudonym-based conditional anonymity in the proposed scheme. Only authorized entities can grant users pseudonyms based on their real identities, which means that the real identities of users can still be revealed by the authorities when necessary.

In response to the above issues, we propose a personalized multi-receiver CLSC scheme with cross-domain authentication for IoV. In our scheme, vehicles can share customized messages with specific receivers in a one-to-many mode. Each receiver can only decrypt the message meant for them, achieving privacy-preserving personalized multi-receiver data sharing. Additionally, we assign a unique tag to each GD to facilitate lightweight cross-domain authentication for dynamic vehicles. To protect the identity privacy of vehicles, we incorporate a conditional privacy method, balancing privacy with the ability to trace malicious activities.

B. Contributions

The contributions of this paper are summarized as follows.

- We propose a lightweight multi-receiver CLSC algorithm that supports privacy-preserving personalized data sharing for IoV. In addition, we present aggregated unsigncryption to further enhance scheme performance.
- We address the dynamic nature of vehicles by enabling efficient cross-domain authentication.
- We introduce a pseudonym generation scheme to achieve conditional anonymity for vehicles.
- We conduct a thorough security analysis and performance assessment, validating the robust security and high efficiency of the proposed scheme. Furthermore, we empirically validate the practical viability of the proposed scheme using a simulated ITS scenario.

C. Organization

The rest of this paper is organized as follows. Section II introduces the related work. Section III presents problem statement, including security assumption, system model, security model, and design goals. Section IV details the construction of our proposed scheme. In Section V, we provide a formal proof of the security properties achieved by our scheme. In Section VI, performance evaluation of our proposed scheme is discussed. Finally, Section VII concludes the paper.

II. Related Work

In this section, we review the related works concerning multi-receiver data sharing and cross-domain authentication, respectively.

A. Multi-receiver Data Sharing

To secure multi-receiver data sharing, the existing schemes often rely on CA or IBC. CA is widely utilized for authentication in multi-receiver data sharing scenarios [23]–[25]. For example, Yang *et al.* [25] proposed a signcryption scheme based on CA to secure wireless communication in the VANET. With the online/offline signcryption mechanism that they adopted, the computational efficiency in their scheme is greatly improved. However, their scheme cannot handle the additional cost for digital certificate management. IBC is another common approach in many multi-receiver data sharing schemes [15], [26]. For instance, Wang *et al.* [15] designed a multi-receiver scheme for IoMT using signcryption, enabling secure data sharing between patients and doctors. In their work, keys are generated based on users' attributes using bilinear pairing, leading to significant computational overheads. In addition, a centralized key generation infrastructure generates keys for all participants, making the system vulnerable to internal attacks. As an innovative technique, Certificateless Public Key Cryptography (CL-PKC) avoids the drawbacks of both CA and IBC, making it a prevalent choice in recent research [16], [27]–[29]. For example, Xu *et al.* [29] proposed a multi-receiver CLSC scheme to safeguard general data transmission. In their design, the system allows users to send multi-receiver messages without worrying the additional

TABLE I
FUNCTIONALITY COMPARISON

Functionality	[11]	[16]	[18]	[25]	[29]	[30]	[31]	[32]	[33]	[34]	Ours
Data Confidentiality	✔	✔	✖	✔	✔	✖	✔	✔	✔	✖	✔
Cross-domain Authentication	✖	✖	✔	✖	✖	✔	✖	✔	✖	✔	✔
Personalized Multi-receiver Data Sharing	✖	✔	✖	✖	✖	✖	✖	✖	✖	✖	✔
Conditional Anonymity	✖	✖	✖	✔	✖	✖	✔	✔	✖	✔	✔
Key Escrow Resistance	✔	✔	✖	✖	✔	✖	✖	✖	✔	✖	✔
Batch Verification	✖	✖	✖	✔	✖	✔	✔	✖	✖	✖	✔

cost from managing certificates. Similarly, Zhou *et al.* [16] proposed another multi-receiver CLSC scheme that, according to formal analysis and experiment results, outperforms many other schemes in both security and efficiency.

It is obvious that the traditional techniques based on CA and IBC can be prohibitively expensive and impose heavy burdens on resource-constrained vehicles. Given the advantages of CL-PKC, CLSC is a much better option for multi-receiver data sharing in IoV. However, existing CLSC-based schemes can be further optimized by removing pairing operations [16]. Moreover, current privacy-perserving schemes only achieve partial anonymity, where either the sender or the receiver is anonymous [16]. This limitation could be addressed by ensuring mutual anonymity in data sharing.

B. Cross-domain Authentication

To enhance the overall security of IoV systems, cross-domain authentication is essential. In recent years, numerous cross-domain authentication methods have been proposed [18], [35]–[37]. Among them, blockchain technology is frequently employed. For instance, Wang *et al.* [35] designed a cross-domain authentication scheme using blockchain to allow all base stations to jointly manage public keys and registration parameters, which effectively reduces the communication overhead between users and base stations. Similarly, Wang *et al.* [36] proposed another cross-domain authentication scheme based on primary–secondary blockchain where the primary chain and the secondary chain are respectively responsible for the transmission of trust between domains. Through the cooperation of the two chains, cross-domain authentication is realized. Chen *et al.* [30] also leveraged blockchain in their scheme. However, these schemes [30], [35] rely on bilinear mapping, leading to high computational overhead unsuitable for IoV devices. To address this, Zhang *et al.* [18] proposed an ECC-based cross-domain authentication scheme without bilinear pairing, reducing computational overhead but introducing additional costs due to its complicated blockchain CA mechanism. To alleviate certificate management burdens, CL-PKC has gained significant attention [36], [38]–[40]. Feng *et al.* [38] deployed a consortium blockchain for cross-domain authentication in the Internet of Industrial Things (IIoT). In this scheme, signatures are generated using CL-PKC and verified through on-chain smart contracts, ensuring both security and efficiency. Similarly, Miao *et al.* [39] utilized a consortium blockchain for cross-domain authentication in vehicle-to-grid networks.

While blockchain is highly regarded for cross-domain authentication, it can limit system scalability due to rapidly growing storage requirements and longer consensus latency. Therefore, only essential data should be stored on-chain to mitigate these issues. Additionally, few works attempted to incorporate CLSC into cross-domain authentication [22]. In summary, the challenge of securing massive and frequent multi-receiver data sharing while enabling flexible and lightweight cross-domain authentication in IoV remains significant and unresolved. To address these gaps, we propose a secure and efficient personalized multi-receiver data sharing scheme with cross-domain authentication for IoV, based on CLSC. Our scheme introduces aggregated unsigncryption to enhance system performance and incorporates conditional privacy protection for users' identity information.

C. Functionality Analysis

We conduct a functionality comparison among the proposed scheme and several related schemes, examining six significant aspects. The comparison results are illustrated in Table I. The objective of the proposed scheme is to achieve these features simultaneously while accommodating the real-time and resource-constrained nature of IoV.

III. Problem Statement

In this section, we present the security assumptions, system model, security model, and design goals of our proposed scheme. The notations are presented in Table II.

TABLE II
Symbol Representation

Symbol	**Definition**
λ	A security parameter
$params$	System public parameters
q	A large prime number
s	Master secret key
$\mathbb{F}_q$	A finite field with order q
E	An elliptic curve
G	An additive cyclic group
$H_1 - H_5$	One-way hash functions
ID_s/ID_r	Identity of sender/receiver
ID_i	Identity of a user
RID_i/PID_i	Real/pseudo identity of a user
PK_i/sk_i	Complete public/secret key of a user
$\mathcal{A}_1^i/\mathcal{A}_2^i$	Type I/II adversaries
$Period_i$	The validity period of PID_i
m_i	Plaintext sent for ID_i
T_i	The current timestamp

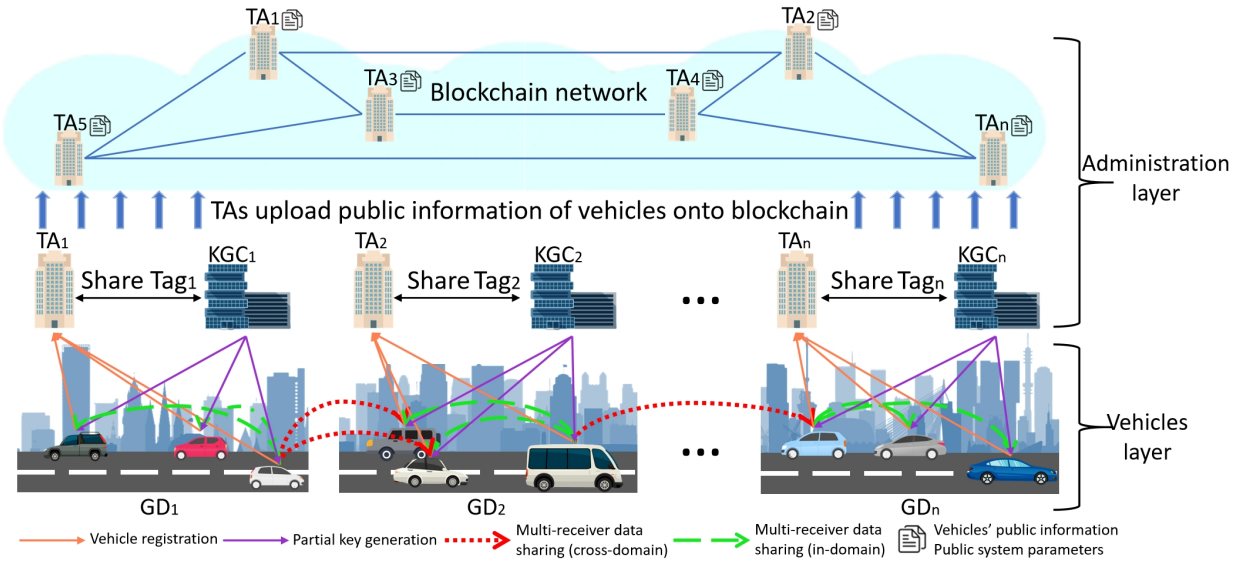

Fig. 1. System model

A. Security Assumptions

The security of the proposed scheme is based on the following two security assumptions.

- *Discrete Logarithm (DL) Assumption*: The DL problem is that given $P, bP \in G$ where P, bP are two points on an elliptic curve and G is a cyclic group, an algorithm needs to calculate the unknown number b. And the DL assumption is that there exists no algorithm that can solve such problems in Probabilistic Polynomial Time (P.P.T) with a non-negligible advantage.
- *Computational Diffie Hellman (CDH) Assumption*: The CDH problem is that given $P, aP, bP \in G$, an algorithm needs to compute abP. And the CDH assumption is that no algorithm can solve this problem within P.P.T with a non-negligible advantage.

B. System Model

As illustrated in Fig 1, the proposed IoV system comprises four types of entities: Key Generation Center (KGC), Trusted Authority (TA), blockchain network, and vehicles. The first three components collectively form the administration layer which maintains reliable operation of the IoV system via parameter generation, vehicle management, and behavior auditing. The heterogeneous vehicles constitute the vehicles layer, facilitating secure and efficient traffic status based on ubiquitous data sharing.

- **KGC**: KGC acts as the system administrator. It initializes the system and is responsible for generating partial public keys, partial private keys, and pseudonymous identities for vehicles. Additionally, the KGC can trace the real identities of malicious entities.
- **TA**: TAs are represented by governments within GDs. Each of them generates a unique tag for its GD. Upon vehicle registration, the TA loads the tag into the vehicles' On-Board Units (OBUs) and uploads the vehicles' identities onto blockchain.
- **Blockchain Network**: A public blockchain, where each TA functions as a full node, serves as a data access platform within the IoV system. TAs from different GDs upload public parameters, vehicle pseudonyms, and corresponding public keys onto this platform. The incorporation of blockchain is due to its advantages of decentralization, transparency, and immutability, which collectively enhance the reliability of the system.
- **Vehicles**: Each vehicle is equipped with an OBU and various sensors. An OBU is a tamper-resistant trusted execution environment (TEE), typically implemented on trusted hardware platforms, e.g., ARM TrustZone [41]. It can ensure the confidentiality and integrity of stored information. Vehicles exchange traffic data wirelessly through Dedicated Short-Range Communications (DSRC) or C-V2X technology.

C. Security Model

Generally, in a CLSC scheme, there are two types of adversaries [16], [33]: Type-I adversaries, who are external attackers capable of replacing a user's public key with another known to them, and Type-II adversaries, who are internal attackers with knowledge of the master secret key. Let $\mathcal{A}_1^i$ represent Type-I adversaries and $\mathcal{A}_2^i$ represent Type-II adversaries. With these two types of adversaries, a CLSC scheme must ensure two security properties: confidentiality and unforgeability. Let $\mathcal{A}_1^1$ and $\mathcal{A}_2^1$ target confidentiality, while $\mathcal{A}_1^2$ and $\mathcal{A}_2^2$ target unforgeability.

- **Confidentiality**: To verify whether a CLSC scheme can maintain the confidentiality of users' messages, specific interactive games are designed and executed. In a typical game, $\mathcal{A}_1^1$ and $\mathcal{A}_2^1$ send oracle queries to a challenger $\mathcal{C}$ and use the responses to deduce the scheme's functionality. After a series of queries, they receive a ciphertext and two plaintexts. Their task is to guess which plaintext corresponds to the ciphertext. If they cannot solve

this with a non-negligible advantage, the CLSC scheme is considered *indistinguishable under adaptively chosen ciphertext attacks* (IND-CLSC-CCA2).

- **Unforgeability**: Similarly, to verify the unforgeability of the scheme, $\mathcal{A}_1^2$ and $\mathcal{A}_2^2$ engage in another interactive game with challenger $\mathcal{C}$ by making oracle queries. If they cannot forge a ciphertext that can be accepted by a receiver with a non-negligible advantage, the CLSC scheme is considered to have *existential unforgeability under adaptive chosen message attacks* (EUF-CLSC-CMA).

D. Design Goals

Based on the system model and security model outlined above, our scheme aims to achieve the following objectives:

- **Flexibility**: Data sharing scenarios in IoV can be highly complicated. Therefore, a scheme that supports not only one-to-one and many-to-one transmission but also personalized multi-receiver data sharing is essential.
- **Confidentiality**: Traffic data is inherently sensitive and private. Thus, messages must be encrypted before transmission, ensuring that only the intended recipients can decrypt the ciphertexts.
- **Unforgeability**: Preventing the forgery of ciphertexts with false road information is crucial not only for individual driver safety but also for the overall efficiency of the traffic system.
- **Low Computational Cost**: IoV devices typically have limited resources. It is necessary to design protocols that minimize computational overhead.
- **Conditional Anonymity**: Using pseudonyms to mask users' true identities effectively protects their privacy. However, system administrators must have the ability to trace the real identity of any malicious entity.
- **Cross-Domain Authentication**: Cross-domain adversaries can severely disrupt traffic. Hence, establishing a lightweight cross-domain authentication scheme is imperative for ensuring continuous and reliable IoV services.

IV. Scheme Design

In this section, we describe the detailed construction of the proposed scheme, which is composed of six phases. The processes are outlined in Fig 2 and Fig 3.

A. Setup

This phase aims to initialize the system.

1) KGC first generates an elliptic curve E on the finite field $\mathbb{F}_q$ whose order is q and generator is P.
2) TA randomly chooses a secret $Tag \in Z_q^*$ for its GD. Note that Tag will be stored confidentially in a tamper-resistant OBU installed on each vehicle registered in this GD and it will not be accessible to the owner of the vehicle and cannot be extracted. KGC receives this Tag from TA through an offline channel. Then KGC computes $TAG = Tag \cdot P$ and publicizes the result in the system by uploading it to the blockchain.
3) KGC randomly selects the master secret key $s \in Z_q^*$ and computes $P_{pub} = s \cdot P$ as the public key.
4) KGC continues to define a cyclic group G on E and defines five one-way hash functions as follows:
   - $H_1 : \{0,1\}^* \times G \times G \to \{0,1\}^l$
   - $H_2 : \{0,1\}^* \times G \times G \times G \to Z_q^*$
   - $H_3 : \{0,1\}^l \times G \times G \to \{0,1\}^k$
   - $H_4 : \{0,1\}^k \times G \times G \to \{0,1\}^k$
   - $H_5 : \{0,1\}^l \times G \times G \times \{0,1\}^k \times G \times \{0,1\}^* \to Z_q^*$

   In these hash functions, l denotes the length of an identity and k denotes the length of a message.
5) Finally, KGC uploads the system parameter $params = (G, q, P, P_{pub}, H_1, H_2, H_3, H_4, H_5)$ to the blockchain.

B. Partial Key Generation

This algorithm is executed by the KGC when a user (or vehicle) with ID_s requests for partial keys and a pseudonym.

1) First, the user ID_s whose real identity is RID_s chooses a secret key $x_s \in Z_q^*$ randomly and calculates the first partial public key $X_s = x_s \cdot P$. The user then sends $\{RID_s, X_s\}$ to KGC.
2) With the data received, KGC computes $h_s^1 = H_1(Period_s, P_{pub}, s \cdot X_s)$ where $Period_s$ is a binary string representing the validity period of the generated pseudonym. KGC then generates a pseudonym via computing $PID_s = RID_s \oplus h_s^1$.
3) Next, KGC randomly selects the number $a_s \in Z_q^*$ and computes the partial public key for ID_s through calculating $A_s = a_s \cdot P$.
4) To generate a partial secret key, KGC first computes $h_s^2 = H_2(PID_s, X_s, A_s, P_{pub})$. Subsequently, KGC computes the partial secret key $y_s = a_s + h_s^2 \cdot s$.
5) Finally, KGC sends $\{PID_s, Period_s, A_s, y_s, T_0\}$ to ID_s through a secure channel (or in the offline mode), where T_0 is a timestamp. Also, the co-relation between ID_s's pseudonym PID_s and complete public key $PK_s = \{X_s, A_s\}$ will be uploaded to the blockchain at the same time.

Notably, only the KGC, possessing the master secret key s, can compute h_s^1 and generate valid pseudonyms. Consequently, KGC is able to determine the real identity RID_s by efficiently calculating $RID_s = PID_s \oplus h_s^1$. This mechanism illustrates how our proposed scheme achieves conditional anonymity for privacy-sensitive vehicles. Also, this pseudonym is only for covering up the real identity of a user. It cannot protect one's location privacy. For those interested in location privacy techniques, this survey serves as a valuable resource [42]. In addition, KGCs can design a vehicle revocation scheme by creating a revocation list on the blockchain to record the expired public keys [43].

C. Complete Key Generation

This algorithm is executed by users after they receive partial keys from the KGC.

1) First, ID_s checks the freshness of T_0. If it is out-dated, the user needs to send requests to KGC again.

2) If T_0 is valid, ID_s computes:
   - $h_s^{1'} = H_1\left(Period_s, P_{pub}, P_{pub} \cdot x_s\right)$
   - $PID_s^{'} = RID_s \oplus h_s^{1'}$

   After that, ID_s verifies if $PID_s = PID_s^{'}$. If it does, PID_s is valid, otherwise ID_s needs to send requests again.
3) Then, ID_s computes $h_s^{2'} = H_2\left(PID_s, X_s, A_s, P_{pub}\right)$ and checks if $y_s \cdot P = A_s + h_s^{2'} \cdot P_{pub}$. If it does not, ID_s sends requests again. Otherwise, ID_s accepts PID_s as its pseudo identity, A_s as its partial public key, and y_s as its partial secret key.
4) After the above steps, $\{X_s, A_s\}$ is set as ID_s's complete public key and $\{x_s, y_s\}$ is the complete private key.

Note that if the request is sent again, KGC will generate both new pseudonym and partial keys for ID_s and upload the pseudonym and public key to blockchain again.

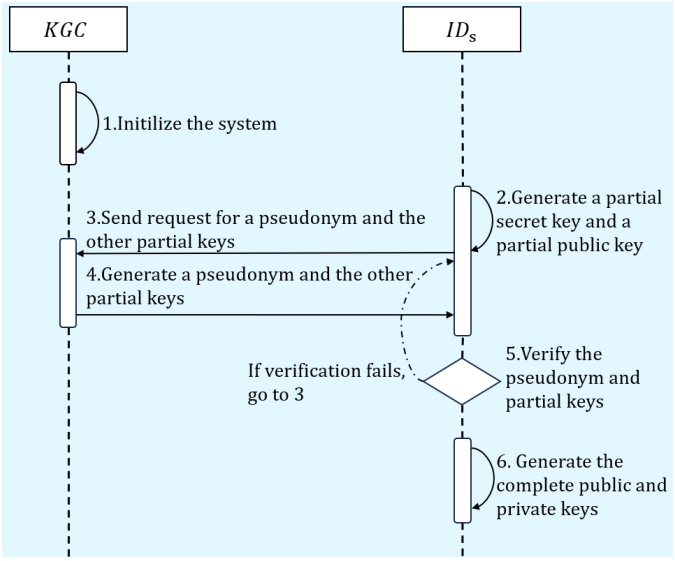

Fig. 2. Workflow of *Setup*, *Partial Key Generation*, and *Complete Key Generation*

To sum up, in real-world IoV scenarios, the complete process of the keys generation in our proposed scheme is as follows. First, when a person buys a vehicle, he/she must register to the local TA which can be a vehicle management office. The workers there will load the Tag corresponding to the GD into the OBU of the vehicle offline. Then the user generates partial public and secret keys and registers to the local KGC. In return, the KGC generates the other partial keys and a pseudonym for the user. In addition, the KGC publicizes the correlation between the pseudonym and the public keys after generating them. When the user receives the KGC's response, he/she will verify the validity of the pseudonym and the other partial keys. If the verification fails, the user will register to the KGC again. Otherwise, the user can use the keys and Tag to signcrypt data and perform cross-domain authentication from then on.

D. Personalized Multi-receiver Signcryption

This algorithm is executed by a data sender ID_s when he/she has n customized messages intended for n specific receivers.

1) ID_s randomly selects $u \in Z_q^*$ and computes $U = u \cdot P$.
2) For each $i \in [1, n]$, ID_s performs the following calculations one by one:
   - $h_i^2 = H_2\left(PID_i, X_i, A_i, P_{pub}\right)$
   - $W_i = u \cdot \left(X_i + A_i + h_i^2 \cdot P_{pub}\right)$
   - $B_i = H_3\left(PID_i, U, W_i\right)$
   - $c_i = B_i \oplus m_i$

   Note that each PID_i and $PK_i = \{X_i, A_i\}$ are found on the blockchain.
3) Afterwards, ID_s creates the shared ciphertext $C = \{H_4\left(B_1, TAG, W_1\right) \parallel c_1, \ldots, H_4\left(B_n, TAG, W_n\right) \parallel c_n\}$.
4) Subsequntly, ID_s computes:
   - $h^s = H_5\left(PID_s, X_s, A_s, C, TAG, T\right)$
   - $\sigma = u + Tag + h^s \cdot \left(x_s + y_s\right)$

   Note that Tag is the unique tag of the GD that ID_s belongs to, which is used for identity authentication.
5) Finally, the data package $\{PID_s, \sigma, C, TAG, U, T\}$, where T is a timestamp, is sent to every receiver.

E. Personalized Multi-receiver Unsigncryption

This algorithm is performed by a receiver ID_i who receives $\{PID_s, \sigma, C, TAG, U, T\}$. It first obtains the public keys $PK_s = \{X_s, A_s\}$ corresponding to PID_s from blockchain. Then ID_i verifies signature and conducts cross-domain authentication. If the message passes the verification, ID_i unsigncrypts the message.

1) ID_i first examines the freshness of T. If it is out-dated, ID_i terminates this process.
2) Next, ID_i conducts the following calculations:
   - $W_i = U \cdot \left(x_i + y_i\right)$
   - $B_i = H_3\left(PID_i, U, W_i\right)$
   - $h^{s'} = H_5\left(PID_s, X_s, A_s, C, TAG, T\right)$
   - $h_s^{2'} = H_2\left(PID_s, X_s, A_s, P_{pub}\right)$
3) Then, ID_i verifies the signature and conducts cross-domain authentication by checking if $\sigma \cdot P = U + TAG + h^{s'} \cdot \left(X_s + A_s + h_s^{2'} \cdot P_{pub}\right)$. If the equation does not hold, the sender either fails in providing a valid signature or using the right Tag. Specifically, an unauthorized vehicle will not obtain a valid Tag to generate a signature that can successfully pass verification, thereby ensuring cross-domain authentication. Therefore, ID_i refuses the ciphertext. Otherwise, the receiver continues.
4) ID_i computes $H_4\left(B_i, TAG, W_i\right)$ and then finds $H_4\left(B_i, TAG, W_i\right) \parallel c_i$ in C to get c_i.
5) In the end, ID_i gets the plaintext $m_i = B_i \oplus c_i$.

The correctness proof of signature verification is demonstrated below.

$$\begin{aligned} \sigma \cdot P &= \left(u + Tag + h^s \cdot \left(x_s + y_s\right)\right) \cdot P \\ &= U + TAG + h^s \cdot \left(x_s + a_s + h_s^2 \cdot s\right) \cdot P \\ &= U + TAG + h^s \cdot \left(X_s + A_s + h_s^2 \cdot P_{pub}\right) \end{aligned} \tag{1}$$

The first line of the equation shows how $\sigma \cdot P$ is calculated using ID_s's secret key. The third line shows how $\sigma \cdot P$ can be calculated using ID_s's public key. If this equations holds, ID_r can be sure that this message is indeed from ID_s because the public key matches the secret key.

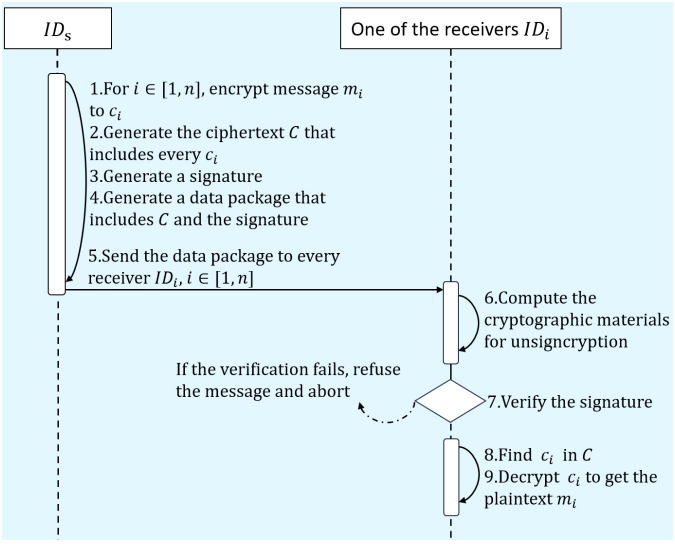

Fig. 3. Workflow of *Multi-receiver Signcryption* and *Multi-receiver Unsigncryption*

F. Extension of Aggregated Unsigncryption

In large cities, especially during rush hours, numerous vehicles might communicate simultaneously, leading to a vehicle receiving multiple messages from different senders within a time slot [20]. Verifying each signature individually under such conditions is highly time-consuming, causing messages to lose their value. Therefore, our proposed scheme allows for batch verification of aggregated signatures, significantly saving vehicles' message processing time.

We assume this algorithm is performed by the receiver ID_r and he/she received n messages at the same time.

1) For each C^i from $PID_i, i \in [1,n]$, after checking the freshness of the message and obtaining the public keys from the blockchain, ID_r performs the following calculations:
   - $W_r^i = U_i \cdot (x_r + y_r)$
   - $B_r^i = H_3(PID_r, U_i, W_r^i)$

   Then ID_i calculates $H_4(B_r^i, TAG_i, W_r^i)$ and finds $H_4(B_r^i, TAG_i, W_r^i) \parallel c_r^i$ in C^i to get c_r^i and computes:
   - $h^{i'} = H_5(PID_i, X_i, A_i, C_r^i, TAG_i, T_i)$
   - $h_i^{2'} = H_2(PID_i, X_i, A_i, P_{pub})$
2) Subsequently, ID_i verifies the aggregated signature by checking if $\left(\sum_{i=1}^n \sigma_r^i\right) \cdot P = \sum_{i=1}^n \left(TAG_i + U_i + h^{i'} \cdot (X_i + A_i + h_i^{2'} \cdot P_{pub})\right)$. If this equation does not hold, ID_i can locate the problematic position using the binary search method. Otherwise, the receiver continues.
3) ID_i computes $m_r^i = B_r^i \oplus c_r^i$ and accepts $m_r^i, i \in [1,n]$.

The correctness of aggregated signature verification is shown below.

$$\begin{aligned}
&\left(\sum_{i=1}^n \sigma_r^i\right) \cdot P \\
=&\left\{\sum_{i=1}^n \left(Tag_i + u_i + h^{i'} \cdot (x_i + y_i)\right)\right\} \cdot P \\
=&\left(\sum_{i=1}^n Tag_i\right) \cdot P + \left(\sum_{i=1}^n u_i\right) \cdot P \\
&+ \left(\sum_{i=1}^n h^{i'} \cdot (x_i + a_i + h_i^{2'} \cdot s)\right) \cdot P \\
=&\sum_{i=1}^n \left(TAG_i + U_i + h^{i'} \cdot (X_i + A_i + h_i^{2'} \cdot P_{pub})\right)
\end{aligned} \tag{2}$$

Algorithm 1: *Aggregated Unsigncryption*

Input: $PID_i, \sigma_r^i, C^i, U_i, TAG_i, T_i, i \in [1,n]$
Output: $m_r^i, i \in [1,n]$

```
for each i ∈ [1,n] do
    Examine T_i
    Compute W_r^i, B_r^i and H_4(B_r^i, TAG_i, W_r^i) to get c_r^i from C^i
    Compute h^{i'} and h_i^{2'}
end
if (Σ_{i=1}^n σ_r^i) · P = Σ_{i=1}^n (TAG_i + U_i) + Σ_{i=1}^n h^{i'} · (X_i + A_i + h_i^{2'} · P_pub) then
    for each i ∈ [1,n] do
        m_r^i ← B_r^i ⊕ c_r^i
    end
else
    refuse the message
end
```

Clearly, this method can greatly improve the performance. If a receiver intends to verify n signatures one by one, they need to perform the operation of point multiplication n times in calculating $\sigma \cdot P$. With this technique, the time-consuming point multiplication operation only needs performing once because the aforementioned operations are replaced by calculating $\left(\sum_{i=1}^n \sigma_r^i\right) \cdot P$. Although this technique adds an extra $3 \cdot n$ point addition operations because many parameters need to be summed up during this verification process, it is still a worthwhile trade-off since point multiplication is much more complicated than point addition.

V. Security Analysis

In this section, we prove that the proposed scheme satisfies the security requirements through the widely-used random oracle model and also perform an informal discussion.

We follow the standard two-notion framework for certificateless signcryption: IND-CLSC-CCA2 confidentiality and EUF-CLSC-CMA unforgeability against Type-I (outsider) and Type-II (insider) adversaries. In the proof of confidentiality, the signcryption algorithm encrypts the messages without signing them, and the unsigncryption algorithm only decrypts the ciphertext. Additionally, some signature information is omitted from the exchanged messages. Similarly, in the proof of unforgeability, the signcryption algorithm only generates signatures for the messages, ignoring the encryption process, and the unsigncryption algorithm only verifies the validity of

the signatures. Consequently, plaintext is exchanged directly instead of ciphertext. To further simplify, pseudo identity is not considered in this section.

A. Confidentiality

Theorem 1 and *Theorem 2* formally establish our scheme's confidentiality against Type-I adversaries $\mathcal{A}_1^1$ and Type-II adversaries $\mathcal{A}_2^1$, respectively.

Theorem 1: If there exists an adversary $\mathcal{A}_1^1$ who can break the confidentiality of our CLSC scheme with a non-negligible advantage ε_1^1 in P.P.T, there exists a challenger $\mathcal{C}$ that is capable of solving a CDH problem instance in P.P.T with a non-negligible advantage $\frac{\varepsilon_1^1}{eq(Q_{sk}+Q_s+Q_u+n)}$, where Q_{sk} is the number of private key extraction queries, Q_s is the number of signcryption queries, Q_u is the number of unsigncryption queries, n is the number of identities in the challenge identity set and e is the base of natural logarithm.

Proof: Given a CDH problem instance $(\alpha P, \beta P, P)$ where $\alpha, \beta \in Z_q^*$, the challenger $\mathcal{C}$ intends to solve it by figuring out $\alpha\beta P$. To achieve this goal, $\mathcal{C}$ creates a simulation environment running our proposed scheme and interacts with the adversary $\mathcal{A}_1^1$ as described below.

1) Setup: The challenger $\mathcal{C}$ performs the *Setup* algorithm to produce system public parameters $params = (G, q, P, P_{pub}, H_1, H_2, H_3, H_4, H_5)$ where $P_{pub} = \beta P$. Then $\mathcal{C}$ sends $params$ to $\mathcal{A}_1^1$. In addition, $\mathcal{C}$ initializes two lists L_Q and L_U to record the queries that the adversary submits. In the former list, public data of users are recorded, while in the latter one, complete data are recorded.

2) Queries Stage: $\mathcal{A}_1^1$ can ask the following queries.

i) *Public Key Extraction Query* : First, $\mathcal{A}_1^1$ sends this query with an identity ID_i to $\mathcal{C}$. Next, $\mathcal{C}$ checks if the tuple $(ID_i, X_i, A_i, x_i, y_i, \eta_i)$ exists in L_U. If it does, $\mathcal{C}$ returns $PK_i = (X_i, A_i)$ to $\mathcal{A}_1^1$. Otherwise, $\mathcal{C}$ randomly chooses $u_i \in Z_q^*$ and $\eta_i \in \{0, 1\}$. Depending on the value of η_i, $\mathcal{C}$ performs following operations.
- If $\eta_i = 0$, $\mathcal{C}$ randomly selects $x_i, a_i, h_i^2 \in Z_q^*$ to compute $X_i = x_i P, A_i = a_i P$ .If $(*, X_i, A_i, *, *, *)$ already exists in L_U, $\mathcal{C}$ randomly chooses x_i and a_i again. $\mathcal{C}$ continues to compute $y_i = a_i + h_i^2 \cdot \beta$. It then adds relevant tuples $(ID_i, X_i, A_i, x_i, y_i, \eta_i)$ into L_U and $\left(ID_i, X_i, A_i, h_i^2, P_{pub}\right)$ into L_Q.
- If $\eta_i = 1$, meaning this user has been compromised, $\mathcal{C}$ chooses $\alpha_1, \alpha_2 \in Z_q^*$ randomly and computes $X_i = \alpha_1 P, A_i = \alpha_2 P$. If $(*, X_i, A_i, *, *, *)$ already exists in L_U, $\mathcal{C}$ chooses two random numbers again. Finally, $\mathcal{C}$ adds relevant tuples $(ID_i, X_i, A_i, *, *, \eta_i)$ and $(ID_i, X_i, A_i, h_2, P_{pub})$ into L_U and L_Q respectively.

Finally, $\mathcal{C}$ sends (X_i, A_i) to $\mathcal{A}_1^1$.

ii) H_2 *Query*: When $\mathcal{A}_1^1$ sends this query to $\mathcal{C}$ with $(ID_i, X_i, A_i, P_{pub})$ as input, $\mathcal{C}$ first checks if $\left(ID_i, X_i, A_i, h_i^2\right)$ exists in L_Q. If it does, $\mathcal{C}$ returns h_i^2 directly to $\mathcal{A}_1^1$. Otherwise, $\mathcal{C}$ performs the *Public Key Extraction Query* with ID_i as input and returns h_i^2 to $\mathcal{A}_1^1$ afterwards.

iii) *Private Key Extraction Query*: When $\mathcal{A}_1^1$ sends this query to $\mathcal{C}$ with an identity ID_i as input, $\mathcal{C}$ searches for the targeted tuple (ID_i, X_i, y_i, η_i) in L_U first. If it does not exist, $\mathcal{C}$ performs the *Public Key Extraction Query* with ID_i as input. Next, $\mathcal{C}$ checks the value of η_i. If $\eta_i = 1$, $\mathcal{C}$ aborts. Otherwise, $\mathcal{C}$ returns (x_i, y_i) to $\mathcal{A}_1^1$.

iv) *Public Key Replacement Query*: Using this query, $\mathcal{A}_1^1$ can replace ID_i's public key PK_i with another public key $PK_i' = (X_i', A_i')$.

v) *Signcryption Query*: When $\mathcal{C}$ receives this query from $\mathcal{A}_1^1$ with the input tuple (ID_S, ID_R, M), where $ID_R = (ID_1, ID_2, \ldots, ID_n)$ and $M = (m_1, m_2, \ldots, m_n)$, $\mathcal{C}$ traverses L_U for $ID_i\,(i \in [1, n])$ to obtain their data tuples $(ID_i, X_i, A_i, x_i, y_i, \eta_i)$. Then, depending on the value of η_i, $\mathcal{C}$ chooses to take the following actions:
- If $\eta_i = 1$, $\mathcal{C}$ aborts.
- Otherwise, $\mathcal{C}$ performs the *Multi-receiver Signcryption* with users' data in L_U and L_Q to generate a message $(ID_S, \sigma, C_r, TAG, U, T)$, where C_r is the one-to-many ciphertext and T is a timestamp. Next, $\mathcal{C}$ send the message to $\mathcal{A}_1^1$.

vi) *Unsigncryption Query*: $\mathcal{A}_1^1$ sends this query to $\mathcal{C}$ with a message $(ID_S, \sigma, C_r, TAG, U, T)$, ID_S,ID_i as input. When $\mathcal{C}$ receives this query, it first searches L_U for the information tuple $(ID_i, X_i, A_i, x_i, y_i, \eta_i)$ and checks η_i to take different actions.
- If $\eta_i = 1$, $\mathcal{C}$ aborts.
- Otherwise, $\mathcal{C}$ runs *Multi-receiver Unsigncryption* algorithm to decrypt the message and sends the plaintext m_i to $\mathcal{A}_1^1$.

3) Challenge: After $\mathcal{A}_1^1$ asks above queries for polynomial time, it sets ID_S as the sender and $ID_R = (ID_1, ID_2, \ldots, ID_n)$ as the receivers set. Then it generates two challenge messages sets i.e. $M_0 = \left(m_0^1, m_0^2, \ldots, m_0^n\right)$ and $M_1 = \left(m_1^1, m_1^2, \ldots, m_1^n\right)$. In both M_0 and M_1, $\forall i \in [1, n]$, $\left|m_0^i\right| = \left|m_1^i\right|$. Next, $\mathcal{C}$ performs *Public Key Extraction Query* for every ID_i in the receivers set, generating their information tuples $(ID_i, X_i, A_i, x_i, y_i, \eta_i)$. After that, depending on the generated data, the following operations will be performed.
- If $\forall i \in [1, n], \eta_i = 0$, $\mathcal{C}$ aborts.
- Otherwise, $\exists j \in [1, n], \eta_j = 1$. $\mathcal{C}$ first sets $U = \alpha P$. Next, $\forall i \in [1, n]$, $i \neq j$, $\mathcal{C}$ computes $W_i = U\,(x_i + y_i)$. Then $\mathcal{C}$ chooses $W_j \in G$. After that, for $\forall i \in [1, n]$, $\mathcal{C}$ computes $B_i = H_3\,(ID_i, U, W_i)$ and $c_i = B_i \oplus m_d^i$, where $d \leftarrow \{0, 1\}$. Subsequently, $\mathcal{C}$ generates a one-to-many ciphertext $C_R = \{H_4\,(B_1, TAG, W_1) \parallel c_1, \ldots, H_4\,(B_n, TAG, W_n) \parallel c_n\}$.

Finally, $\mathcal{C}$ sends this challenge message $\{ID_S, C_R, TAG, U, T\}$ to $\mathcal{A}_1^1$.

4) Guess: $\mathcal{A}_1^1$ needs to guess $d' \in \{0, 1\}$ after it receives $\mathcal{C}$'s message. If $d' = d$, $\mathcal{C}$ outputs $\alpha\beta P = \frac{1}{h_2}\left(W_j - (a_1 + a_2)\,U\right)$, in which $U = \alpha P$, $W_j = \alpha\left(X_j + A_j + h_2\beta P\right)$ and $h_2 = H_2\left(ID_j, X_j, A_j, P_{pub}\right)$, as the solution of the given CDH problem instance.

5) Probability Analysis: The probability of finding the index j, with which $\eta_j = 1$, is $Pr\left[\eta_j = 1\right] = \frac{1}{Q_s + Q_u + Q_{sk} + n} = \delta = Pr[FI]$. The probability of guessing out the true challenge message set is $Pr\left[G\right] = \frac{1}{q}$. The probability of not aborting in query phase is $Pr\left[NAQ\right] = (1 - \delta)^{(Q_s + Q_u + Q_{sk})}$. Thus,

$Pr[G \wedge NAQ \wedge FI] = \frac{\delta(1-\delta)^{(Q_s+Q_u+Q_{sk})}}{q}$.

Assuming that Q_s, Q_{sk} and Q_u are large enough, $Pr[NAQ]$ approaches e^{-1}. So, $Pr[G \wedge NAQ \wedge FI]$ can be denoted as $\frac{1}{eq(Q_s+Q_u+Q_{sk}+n)}$. Therefore, if $\mathcal{A}_1^1$ can break the confidentiality of our proposed CLSC scheme with a non-negligible advantage ε_1^1, it can solve a CDH problem instance with a non-negligible advantage $\frac{\varepsilon_1^1}{eq(Q_s+Q_u+Q_{sk}+n)}$.

Theorem 2: If there exists an adversary $\mathcal{A}_2^1$ who can break the confidentiality of our CLSC scheme with a non-negligible advantage ε_2^1 in P.P.T, there exists a challenger $\mathcal{C}$ that is capable of solving a CDH problem instance in P.P.T with a non-negligible advantage $\frac{\varepsilon_2^1}{eq(Q_{sk}+Q_s+Q_u+n)}$.

Proof: The goal of the challenger $\mathcal{C}$ is to solve a CDH problem instance described below. That is, with the input challenge tuple $(\alpha P, \beta P, P)$, where $\alpha, \beta \in Z_q^*$, $\mathcal{C}$ needs to find out $\alpha\beta P$. In order to accomplish this task, $\mathcal{C}$ creates a simulated environment in which our proposed scheme is performed and interacts with $\mathcal{A}_2^1$. The interactive game between $\mathcal{C}$ and $\mathcal{A}_2^1$ is as follows.

1) Setup: The challenger $\mathcal{C}$ performs the *Setup* algorithm to produce system public parameters $params = (G, q, P, P_{pub}, H_1, H_2, H_3, H_4, H_5)$. Then $\mathcal{C}$ sends $params$ and the master secret key $msk = s$ to $\mathcal{A}_2^1$. In addition, $\mathcal{C}$ initializes a list L_U to record the queries that are submitted by the adversary.

2) Queries Stage: The following two queries together with the signcryption query and the secret key extraction query can be asked by $\mathcal{A}_2^1$. The latter two queries are not introduced below because they are the same as the ones in Theorem 1.

i) Public Key Extraction Query: When $\mathcal{C}$ receives this query from $\mathcal{A}_2^1$ with the input ID_i, it first checks whether the tuple $(ID_i, X_i, A_i, x_i, y_i, \eta_i)$ exists in L_U. If it does, $\mathcal{C}$ directly returns (X_i, A_i) to $\mathcal{A}_2^1$. Otherwise, $\mathcal{C}$ chooses a random value $\eta_i \leftarrow \{0, 1\}$. Next, $\mathcal{C}$ chooses to take the following actions according to η_i.
   - If $\eta_i = 0$, $\mathcal{C}$ randomly selects $x_i, a_i \in Z_q^*$ to compute $X_i = x_i P, A_i = a_i P$ and $h_i^2 = H_2(ID_i, X_i, A_i, P_{pub})$. $\mathcal{C}$ continues to compute $y_i = a_i + h_i^2 \cdot s$. It then adds relevant tuples $(ID_i, X_i, A_i, x_i, y_i, \eta_i)$ into L_U and sends (X_i, A_i) to $\mathcal{A}_2^1$.
   - If $\eta_i = 1$, also meaning this user has been compromised, $\mathcal{C}$ randomly chooses $x_i = \alpha_1 \in Z_q^*$ ensuring that the tuple $(*, *, x_i, *, *)$ does not exist in L_U. Then, $\mathcal{C}$ computes $X_i = \alpha_1 P = x_i P$ and $A_i = \beta P$. After that, $\mathcal{C}$ adds the tuple $(X_i, A_i, x_i, *, \eta_i)$ into L_U. Finally, $\mathcal{C}$ returns (X_i, A_i) to $\mathcal{A}_2^1$.

ii) Unsigncryption Query: When the adversary $\mathcal{A}_2^1$ sends this query along with its required input ID_i and an encrypted message $(ID_S, \sigma, C_r, TAG, U, T)$, $\mathcal{C}$ first traverses L_U for the value of η_i. Accoding to η_i, $\mathcal{C}$ chooses from the following options.
   - If $\eta_i = 1$, $\mathcal{C}$ aborts.
   - Otherwise, $\mathcal{C}$ uses multi-receiver unsigncryption algorithm to decrypt the message and returns the plaintext m_i to $\mathcal{A}_2^1$.

3) Challenge: After $\mathcal{A}_2^1$ asks above queries for polynomial time, it sets ID_S as the sender and $ID_R = (ID_1, ID_2, \ldots, ID_n)$ as the receivers set. Then it generates two challenge messages sets i.e. $M_0 = \left(m_0^1, m_0^2, \ldots, m_0^n\right)$ and $M_1 = \left(m_1^1, m_1^2, \ldots, m_1^n\right)$. In both M_0 and M_1, $\forall i \in [1, n]$, $\left|m_0^i\right| = \left|m_1^i\right|$. Next, $\mathcal{C}$ performs *Public Key Extraction Query* for every ID_i in the receivers set, generating their information tuples $(ID_i, X_i, A_i, x_i, y_i, \eta_i)$. After that, depending on the generated data, the following operations will be performed.

- If $\forall i \in [1, n], \eta_i = 0$, $\mathcal{C}$ aborts.
- Otherwise, $\exists j \in [1, n], \eta_j = 1$. $\mathcal{C}$ first sets $U = \alpha P$. Next, $\forall i \in [1, n]$, $i \neq j$, $\mathcal{C}$ computes $W_i = U(x_i + y_i)$. Then $\mathcal{C}$ chooses $W_j \in G$. After that, for $\forall i \in [1, n]$, $\mathcal{C}$ computes $B_i = H_3(ID_i, U, W_i)$ and $c_i = B_i \oplus m_d^i$, where $d \leftarrow \{0, 1\}$. Subsequently, $\mathcal{C}$ generates a one-to-many ciphertext $C_R = \{H_4(B_1, TAG, W_1) \parallel c_1, \ldots, H_4(B_n, TAG, W_n) \parallel c_n\}$. Finally, $\mathcal{C}$ sends this challenge message $\{ID_S, C_R, TAG, U, T\}$ to $\mathcal{A}_2^1$.

4) Guess: $\mathcal{A}_2^1$ needs to guess $d' \in \{0, 1\}$ after it receives $\mathcal{C}$'s message. If $d' = d$, $\mathcal{C}$ outputs $\alpha\beta P = \frac{1}{h_2}(W_j - (a_1 + sh_2^j)TAG)$, in which $TAG = \alpha P$, $W_j = \alpha(X_j + \beta P + h_2 P_{pub})$ and $h_2 = H_2(ID_j, X_j, A_j, P_{pub})$, as the solution of the given CDH problem instance.

5) Probabilty Analysis: Similar to Theorem 1, if $\mathcal{A}_2^1$ can break the confidentiality of our proposed CLSC scheme with a non-negligible advantage ε_2^1, it can solve a CDH problem instance with a non-negligible advantage $\frac{\varepsilon_2^1}{eq(Q_s+Q_u+Q_{sk}+n)}$.

B. Unforgeability

The unforgeability of our scheme is proven in *Theorem 3* and *Theorem 4*, demonstrating resistance against Type-I adversaries $\mathcal{A}_1^2$ and Type-II adversaries $\mathcal{A}_2^2$, respectively.

Theorem 3: If there exists an adversary $\mathcal{A}_1^2$ who can break the unforgeability of our CLSC scheme with a non-negligible advantage ε_1^2 in P.P.T, there exists a challenger $\mathcal{C}$ that is capable of solving a DL problem instance in P.P.T with a non-negligible advantage $(1 - \frac{1}{e})\frac{\varepsilon_1^2}{e(Q_{sk}+Q_s+Q_u+n)Q_H}$.

Proof: The challenger $\mathcal{C}$ aims to solve the hardness of the DL assumption. That is, given βP and P, $\mathcal{C}$ needs to figure out the secret number $\beta \in Z_q^*$. To tackle this task, $\mathcal{C}$ creates a simulated environment of our scheme, in which it interacts with the adversary $\mathcal{A}_1^2$. The interactive game between $\mathcal{A}_1^2$ and $\mathcal{C}$ is described below.

1) Setup: First, $\mathcal{C}$ sets $P_{pub} = \beta P$. Then $\mathcal{C}$ generates the system public parameters $params = \{G, q, P, P_{pub}, H_1, H_2, H_3, H_4, H_5\}$ which is sent to $\mathcal{A}_1^2$ later. Furthermore, $\mathcal{C}$ initializes two empty lists L_U and L_Q to record the adversary's queries.

2) Query Stage: Some of the queries that $\mathcal{A}_1^2$ can ask are the same as the ones in Theorem 1, including public key extraction, H_2, secret key extraction and public key replacement queries. Thus, they will not be listed below like the others.

i) Signcryption Query: $\mathcal{A}_1^2$ sends this query to $\mathcal{C}$ with ID_S and a message set $M = (m_1, m_2, \ldots, m_n)$ as input. When $\mathcal{C}$ receives it, $\mathcal{C}$ first searches for the information

tuple $(ID_S, X_S, A_S, x_S, y_S, \eta_S)$ in L_U. Then, it takes specific actions according to η_S.

- If $\eta_S = 1$, $\mathcal{C}$ aborts.
- Otherwise, $\mathcal{C}$ performs the multi-receiver signcryption algorithm in our proposed scheme to generate a signature (TAG, U, σ) based on M. Next, $\mathcal{C}$ sends it to $\mathcal{A}_1^2$.

ii) Unsigncryption Query: When this query is sent to $\mathcal{C}$ with the tuple (ID_S, TAG, U, σ) as the corresponding input, $\mathcal{C}$ traverses L_U for the data tuple $(ID_S, X_S, A_S, x_S, y_S, \eta_S)$. Then, it checks the value of η_S and takes specific actions accodingly as follows.

- If $\eta_S = 1$, $\mathcal{C}$ aborts.
- Otherwise, $\mathcal{C}$ performs multi-receiver unsigncryption algorithm to verify the signature and returns the result to $\mathcal{A}_1^2$.

3) Forge: Given the challenge message set (ID_S, M^*), $\mathcal{A}_1^2$ forges a signature (TAG, U, σ_1), where $TAG = Tag \cdot P$, $U = u \cdot P$ and $\sigma_1 = h^S(\alpha_1 + \alpha_2 + \beta h_S^2) + Tag + u$.

If this signature passes the verification and $\eta_S = 1$, $\mathcal{C}$, based on Forking Lemma, replays $\mathcal{A}_1^2$ by changing the output of H_2 Query which is a random oracle. In this way, $\mathcal{A}_1^2$ generates another signature (TAG, U, σ_2) based on the same challenge message set (ID_S, M^*), the same random numbers $\{u, Tag\}$ and a different value $\hat{h}_S^2$ which is the output of H_2 Query. In this new signature, $TAG = Tag \cdot P$, $U = u \cdot P$ and $\sigma_2 = h^S(\alpha_1 + \alpha_2 + \beta\hat{h}_S^2) + Tag + u$. With σ_1 and σ_2, we have

$$\begin{cases} \sigma_1 = h^S(\alpha_1 + \alpha_2 + \beta h_S^2) + Tag + u \\ \sigma_2 = h^S(\alpha_1 + \alpha_2 + \beta \hat{h}_S^2) + Tag + u \end{cases} \tag{3}$$

According to equations above, $\mathcal{C}$ outputs $\beta = \frac{\sigma_1 - \sigma_2}{h^S(h_S^2 - \hat{h}_S^2)}$ as the solution of the DL problem.

4) Probability Analysis: In previous discussion, we know that $Pr[\eta_j = 1] = \frac{1}{Q_s + Q_u + Q_{sk} + n} = \delta$, where n is the size of challenge set. The probability that $\mathcal{C}$ does not abort in query stage is $Pr[NAQ] = (1-\delta)^{(Q_s + Q_u + Q_{sk})}$ and the probability that $\mathcal{C}$ does not abort in forge stage is obviously $Pr[NAF] = \delta$. When Q_s, Q_u and Q_{sk} are large enough, $Pr[NAQ] = \frac{1}{e}$. The probability of $\mathcal{A}_1^2$ succeeding in forging two signatures that can pass the verification $Pr[F]$ is no less than $(1 - \frac{1}{e})\frac{1}{Q_H}$, where Q_H is the number of H_2 Query and $(1 - \frac{1}{e})$ is the approximate probability of succeeding at least once through randomly guessing what the signature can be for countless times.

To sum up, if $\mathcal{A}_1^2$ can break the unforgeability of our proposed scheme with a non-negligible advantage ε_1^2, there exists a challenger $\mathcal{C}$ who can solve a DL problem instance in P.P.T with a non-negligible advantage no less than $(1 - \frac{1}{e})\frac{\varepsilon_1^2}{e(Q_s + Q_u + Q_{sk} + n)Q_H}$.

Theorem 4: If there exists an adversary $\mathcal{A}_2^2$ who can break the unforgeability of our CLSC scheme with a non-negligible advantage ε_2^2 in P.P.T, there exists a challenger $\mathcal{C}$ that is capable of solving a DL problem instance in P.P.T with a non-negligible advantage $(1 - \frac{1}{e})\frac{\varepsilon_1^2}{e(Q_{sk} + Q_s + Q_u + n)Q_H}$.

Proof: The challenger $\mathcal{C}$ aims to solve the hardness of the DL assumption. That is, given βP and P, $\mathcal{C}$ needs to figure out the secret number $\beta \in Z_q^*$. To accomplish this task, $\mathcal{C}$ creates a simulated environment of our scheme and interacts with the adversary $\mathcal{A}_2^2$ in it. The interactive game between $\mathcal{A}_2^2$ and $\mathcal{C}$ is described below.

1) Setup: $\mathcal{C}$ performs the setup algorithm to generate the master secret key msk and the system public parameters $params = (G, q, P, P_{pub}, H_1, H_2, H_3, H_4, H_5)$ in which H_5 is a random oracle. Then the data created are sent to $\mathcal{A}_2^2$. Next, $\mathcal{C}$ initializes two lists i.e. L_Q and L_U to record queries that $\mathcal{A}_2^2$ sent.

2) Query Stage: In this stage, $\mathcal{A}_2^2$ sent public key extraction query and secret key extraction query like *Theorem 2*. The signcryption query and the unsigncryption query are identical with the ones in *Theorem 3*. The remaining query is introduced below.

i) H_5 Query: $\mathcal{A}_2^2$ issues a hash query with $(ID_i, X_i, A_i, C_i, TAG, U, T)$ as its corresponding input. $\mathcal{C}$ first checks L_Q for $(ID_i, X_i, A_i, C_i, TAG, U, h_5^i)$. If this tuple exists, $\mathcal{C}$ returns h_5^i to $\mathcal{A}_2^2$. Otherwise, $\mathcal{C}$ randomly selects $h_5^i \in Z_q^*$ that does not exist in L_Q and adds $(ID_i, X_i, A_i, C_i, TAG, U, h_5^i)$ into L_Q. Finally, h_5^i is returned to $\mathcal{A}_2^2$.

3) Forge: Finally, $\mathcal{A}_2^2$ outputs a forged signature (TAG, U, σ_1) with the challenge message set (ID_S, M^*) under the constraint that queries above have been asked for sufficient times. In this signature, $TAG = Tag \cdot P$, $U = u \cdot P$ and $\sigma_1 = u + Tag + h^S(\alpha_1 + \beta + s \cdot h_S^2)$, where h^S is the output of H_5 Query. If this signature is valid and $\eta_S = 1$, according to Forking lemma, $\mathcal{C}$ replays $\mathcal{A}_2^2$ by altering the output of H_5 Query without changing other parameters. In this way, $\mathcal{A}_2^2$ outputs another signature (TAG, U, σ_2), where $TAG = Tag \cdot P$, $U = u \cdot P$ and $\sigma_2 = Tag + u + \hat{h}^S(\alpha_1 + \beta + s \cdot h_S^2)$. With σ_1 and σ_2, we have

$$\begin{cases} \sigma_1 = h^S(\alpha_1 + \beta + s \cdot h_S^2) + Tag + u \\ \sigma_2 = \hat{h}^S(\alpha_1 + \beta + s \cdot \hat{h}_S^2) + Tag + u \end{cases} \tag{4}$$

According to the equations above, $\mathcal{C}$ outputs $\beta = \frac{\sigma_1 - \sigma_2}{h^S - \hat{h}^S} - \alpha_1 - s \cdot h_S^2$ as the solution of the given DL problem instance.

4) Probability Analysis: Similar to Theorem 3, if $\mathcal{A}_2^2$ can break the unforgeability of our proposed scheme with a non-negligible advantage ε_2^2, there exists a challenger $\mathcal{C}$ who can solve a DL problem instance in P.P.T with a non-negligible advantage no less than $(1 - \frac{1}{e})\frac{\varepsilon_2^2}{e(Q_s + Q_u + Q_{sk} + n)Q'_H}$, where Q'_H is the number of H_5 Query.

C. Informal Security Analysis

- **Conditional Anonymity**: The real identity of vehicles is anonymized by the KGC during registration. Furthermore, if a malicious vehicle is detected, KGC can easily reveal its real identity using $RID_s = PID_s \oplus h_s^1$ for appropriate punitive actions.
- **Cross-Domain Authentication**: We use a secret tag Tag to mark vehicles that have been legally registered in a specific GD. A fake or illegal vehicle will fail the signature verification $\sigma \cdot P = U + TAG + h^{s'} \cdot (X_s + A_s + h_s^{2'} \cdot P_{pub})$ by the receiver, thus enabling lightweight cross-domain authentication.

VI. Performance Evaluation

In this section, we evaluate the performance of our proposed scheme through a comprehensive series of experiments. Furthermore, the implementation codes of the personalized multi-receiver CLSC we proposed have been open-sourced and can be accessed at https://github.com/Cyning7357/Multi-receiver-CLSC. Note that the selection of the blockchain network is flexible; however, a permissioned blockchain framework such as *Hyperledger Fabric* is recommended because it offers private channels between participants, fine-grained access control that matches the administrative structure, and consensus mechanisms designed for known participants. Furthermore, with the help of *Fast Fabric*, an optimization strategy for *Hyperledger Fabric*, the theoretical TPS capacities of this chain exceed 20,000, making it even more competent. For the sake of simplicity, we omit a detailed evaluation and the implementation codes of the blockchain component.

A. Experimental Settings

1) *Environmental Setup:* To simulate the KGC, our experiments are conducted on a laptop equipped with an Intel Core i5-13450HX CPU @ 4.6 GHz, 32 GB RAM, and Ubuntu 20.04 desktop 64-bit, installed on VMware Workstation 17 Pro within a Windows 10 environment. For simulating an OBU on a $Vehicle$, a Raspberry Pi 4 B model is utilized, running Ubuntu 22.04 with a Cortex-A72 (ARM v8) 1.5-GHz CPU and 4 GB RAM. The cryptographic algorithms are implemented using the Charm-Crypto 0.50, pycryptodome and ECDSA cryptographic library in Python. The results are the average values obtained from 1000 repeated experiments.
2) *Benchmark Schemes:* We select four data sharing schemes based on signcryption for comparison against our scheme in terms of computation and communication costs.
   - **CP-CPPHSC** [31]: This work proposed a one-to-one data sharing scheme based on bilinear pairing to secure heterogeneous vehicular communications. It is chosen to demonstrate the performance of an ordinary signcryption scheme with certificates.
   - **Zhang *et al.*** [11]: This work designed another one-to-one data sharing scheme based on CLSC without bilinear pairing for securing data transmission in IoMT.
   - **MCLS** [29]: This work proposed a multi-receiver signcryption scheme based on bilinear pairing. It is selected to demonstrate the performance of a typical multicast data sharing scheme.
   - **RCB-BSC** [23]: This work put forward a pairing-free broadcast signcryption scheme. It sends single message to multiple users. It is chosen to show the performance of an efficient multicast data sharing scheme.
3) *Parameter Settings:* In the experiments, the secp256k1 elliptic curve for ECC-based schemes, pairing group MNT224 for pairing-based schemes, and SHA-256 hash function are adopted. The parameter sizes that are used for calculating communication costs are listed as follows. The sizes of points on the elliptic curve E such as TAG and A_i are set to 40 $bytes$, denoted as P in the subsequent theoretical analysis. The sizes of timestamps T, $Period_s$ and T_0 are set to 4 $bytes$, denoted as T. The sizes of ID_i and PID_i are set to 4 $bytes$, denoted as ID. The sizes of the output of hash function H_2 and partial secret keys such as x_i are set to 20 $bytes$, denoted as s. For simplicity, the size of a message for one user is also set to 20 $bytes$, denoted as S_m. As for schemes based on bilinear pairing, the size of an element in a cyclic group is set to 65 $bytes$, denoted as G.

TABLE III
EXECUTION TIME OF CRYPTOGRAPHIC OPERATIONS

Operation	Runtime on KGC (ms)	Runtime on Vehicles (ms)
TM_{bp}	2.481	15.270
TM_{pm}^{bp}	3.181	16.814
TM_{pa}^{bp}	0.0119	0.0605
TM_{ep}^{bp}	3.183	16.228
TM_{pm}^{ecc}	0.233	2.472
TM_{pa}^{ecc}	0.000905	0.0440
TM_h	0.000303	0.00817
TM_{htp}	0.333	2.416
TM_{ep}	0.0113	0.0954

B. Numerical Evaluation

1) Computational Overhead: The computational overheads are mainly measured by the execution time of different operations in the schemes. The less execution time, the better the scheme's performance. For convenience, we define a few notations for the cryptographic operations used in the schemes, which are demonstrated as follows. TM_{bp}: the execution time of a bilinear pairing operation; TM_{pm}^{bp}: the execution time of a point multiplication operation related to bilinear pairing; TM_{pa}^{bp}: the execution time of a point addition operation related to bilinear pairing; TM_{ep}^{bp}: the execution time of an exponentiation operation in bilinear pairing; TM_{pm}^{ecc}: the execution time of a point multiplication operation related to ECC; TM_{pa}^{ecc}: the execution time of a point addition operation related to ECC; TM_h: the execution time of a general hash function; TM_{htp}: the execution time of a hash-to-point operation on an elliptic curve; TM_{ep}: the execution time of a general exponentiation operation of large integers. We test the execution time of these operations, as listed in Table III.

The analysis of computational overhead is divided into two sides, namely KGC Side and $Vehicle$ Side, each of which includes two processes. As shown in Table IV, PKG stands for **P**artial **K**ey **G**eneration, where KGC generates partial keys for a user. It might also include generation of pseudo identities depending on the specific design of a scheme. SC stands for **S**ign**C**ryption. In this operation, a sender signcrypts n messages for n receivers and creates one or n data packages with the messages and other necessary information. USC stands for **U**n**S**ign**C**ryption. A user unsigncrypts one message sent to them during this operation. For the benchmark [23] that does not adopt CL-PKC, KG stands for **K**ey **G**eneration. Obviously, KGC generates complete key for a user in this process.

- **CP-CPPHSC** [31]: In this work, the signcryption scheme is based on bilinear pairing. The time for a KGC to

perform process PKG for each user is $2{\cdot}TM_{pm}^{bp} \approx 6.362$ ms. As for process SC and USC, the time consumption is $n \cdot TM_{ep}^{bp} + 2 \cdot n \cdot TM_{pm}^{bp} \approx 49.856 \cdot n$ ms and $2 \cdot TM_{bp} + TM_{pm}^{bp} \approx 47.354$ ms respectively. Any of these operations is terribly time-consuming. Apparently, bilinear pairing is too complicated for both real-time traffic and resources-limited IoV devices.

- **Zhang *et al.*** [11]: This paper puts forward another one-to-one CLSC scheme. In their work, KGC spends $TM_{pm}^{ecc} \approx 0.233$ ms on PKG. At the same time, users spend $2 \cdot n \cdot TM_{pa}^{ecc} + (1 + 2 \cdot n) \cdot TM_{pm}^{ecc} \approx 17.392 \cdot n$ ms and $3 \cdot TM_{pa}^{ecc} + 6 \cdot TM_{pm}^{ecc} \approx 14.964$ ms on SC and USC respectively.
- **MCLS** [29]: In this multicasting scheme, to signcrypt and unsigncrypt a message, it takes a user $n \cdot TM_{bp} + n \cdot TM_{ep} + (1 + 2 \cdot n) \cdot TM_{pm}^{bp} \approx 16.814 + 48.993 \cdot n$ ms and $2{\cdot}TM_{pm}^{bp} + TM_{bp} \approx 48.898$ ms respectively. As for KGC, it only needs to spend $TM_{pm}^{bp} \approx 3.181$ ms to generate partial key for each user.
- **RCB-BSC** [23]: In this scheme, it only takes $2{\cdot}TM_{pm}^{ecc} \approx 0.466$ ms for KGC to generate a key and a certificate for a user. In process SC, The total execution time for SC is $n \cdot TM_{htp}^{ecc} + (1 + 3 \cdot n) \cdot TM_{pm}^{ecc} \approx 9.832 \cdot n + 2.472$ ms. However, for USC, the time consumption is $TM_{htp} + (n+2) \cdot TM_{pm}^{ecc} \approx 2.472 \cdot n + 7.360$ ms, which is surprisingly related to the number of receivers.
- **Ours**: In our proposed scheme, it takes KGC $2 \cdot TM_{pm}^{ecc} \approx 0.466$ ms to finish PKG. As for process SC and USC, the time needed is $2 \cdot n \cdot TM_{pa}^{ecc} + (2 \cdot n + 1) \cdot TM_{pm}^{ecc} \approx 5.032 \cdot n + 2.472$ ms and $4 \cdot TM_{pa}^{ecc} + 4 \cdot TM_{pm}^{ecc} \approx 10.064$ ms respectively.

Based on the data in Table III, Fig. 4 shows the comparison of all five schemes' performance in signcryption on $Vehicles$ when the number of receivers increases and Fig. 5 shows the comparison of computational overhead of unsigncryption. For simplicity, we set the number of receivers of each message in USC of [23] to 10.

In PKG or KG on KGC side, our scheme takes $2 \cdot TM_{pm}^{ecc} \approx 0.466$ ms per user, which takes twice as long as the time in [11]. That is because we intend to achieve conditional anonymity, which requires more operations for KGC. However, this additional time is negligible, as it only adds a mere 0.233 ms compared to their solution. Meanwhile, it takes 6.362 ms in [31] and 3.181 ms in [29] to generate partial keys for a user, which are 1365.23% and 682.62% of ours, proving that our scheme is of high efficiency.

Regarding processes SC and USC on vehicle side, clearly, our proposed scheme outperforms all other benchmark schemes [11], [23], [29], [31] with evident advantage. For example, to signcrypt a data package for 100 receivers, the cost in our scheme is approximately 503.2 ms while the costs are 4985.6 ms, 4916.1 ms, 1739.2 ms, and 985.7 ms in [31], [29], [11], and [23], respectively. Apparently, even the most efficient one [23] consumes close to twice the duration compared to our scheme. Likewise, in unsigncrypting a data package for 10 receivers, our time consumption is only 20.5%, 22.1%, 31,4%, and 67.2% of the consumption in [29], [31], [11], and [23], respectively. These results clearly prove that our scheme enjoys prominent efficiency. This phenomenon can be attributed to the following contributing factors. First, our scheme is based on ECC instead of bilinear pairing in [31] and [29], which saves plenty of time. Second, we optimize our scheme for multi-receiver scenarios so that repeated operations can be avoided. Therefore, it outperforms [11] which is a scheme for one-to-one data sharing. Last but not least, we further optimize our scheme by replacing complex hash-to-point operations with easier cryptographic operations in ECC. As a result, our scheme demonstrates much better performance than [23] although it is another scheme that is tailored for multi-receiver data sharing based on ECC.

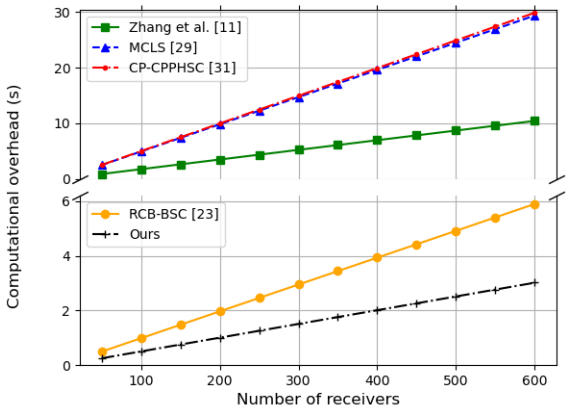

Fig. 4. Computational overhead comparison of signcryption

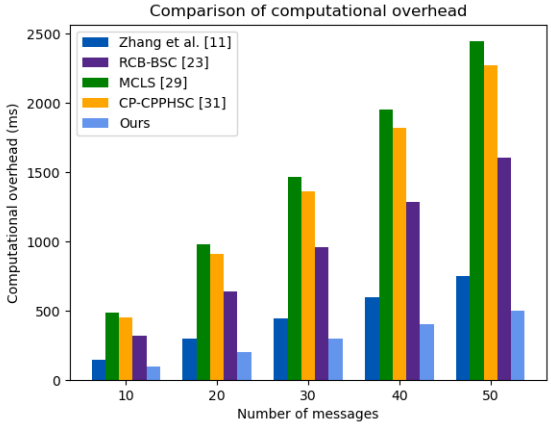

Fig. 5. Computational overhead comparison of unsigncryption

2) Communication Overhead: In this section, we compare the communication overhead of each scheme by analyzing the sizes of the data packages transmitted by users when communicating with multiple receivers. The comparative results are presented in Table V.

Based on the aforementioned parameter setting, we analyze each scheme's communication cost when they are used to send n heterogeneous messages to n receivers.

TABLE IV
COMPARISON OF COMPUTATIONAL OVERHEAD

Scheme	*KGC Side*		*Vehicle Side*	
	PKG	KG	SC	USC
[31]	$2 \cdot TM_{pm}^{bp}$	-	$n \cdot TM_{ep}^{bp} + 2 \cdot n \cdot TM_{pm}^{bp}$	$2 \cdot TM_{bp} + TM_{pm}^{bp}$
[11]	TM_{pm}^{ecc}	-	$2 \cdot n \cdot TM_{pa}^{ecc} + 7 \cdot n \cdot TM_{pm}^{ecc}$	$3 \cdot TM_{pa}^{ecc} + 6 \cdot TM_{pm}^{ecc}$
[29]	TM_{pm}^{bp}	-	$n \cdot TM_{bp} + n \cdot TM_{ep} + (1 + 2 \cdot n) \cdot TM_{pm}^{bp}$	$2 \cdot TM_{pm}^{bp} + TM_{bp}$
[23]	-	$2 \cdot TM_{pm}^{ecc}$	$n \cdot TM_{htp}^{ecc} + (1 + 3 \cdot n) \cdot TM_{pm}^{ecc}$	$TM_{htp} + (2 + n) \cdot TM_{pm}^{ecc}$
Ours	$2 \cdot TM_{pm}^{ecc}$	-	$2 \cdot n \cdot TM_{pa}^{ecc} + (1 + 2 \cdot n) \cdot TM_{pm}^{ecc}$	$4 \cdot TM_{pa}^{ecc} + 4 \cdot TM_{pm}^{ecc}$

TABLE V
COMPARISON OF COMMUNICATION COST

Scheme	PKG	KG	SC
[31]	$2 \cdot G + s$ *bytes*	-	$n \cdot (2 \cdot G + T + S_m)$ *bytes*
[11]	$P + s$ *bytes*	-	$n \cdot (P + s + S_m)$ *bytes*
[29]	G *bytes*	-	$G + (2 \cdot n + 2) \cdot s + n \cdot S_m$ *bytes*
[23]	-	$P + s$ *bytes*	$(n + 1) \cdot P + 2 \cdot s + n \cdot S_m$ *bytes*
Ours	$2 \cdot P + 2 \cdot T + ID + s$ *bytes*	-	$2 \cdot P + T + s + ID + 2 \cdot n \cdot S_m$ *bytes*

- **CP-CPPHSC** [31]: In this paper, KGC generates both pseudo identity and partial secret key to a user in process PKG. Thus, the size of the data package is $2 \cdot G + s = 2 \times 65 + 20 = 150$ $bytes$. In process SC, the sender sends two elements in cyclic groups and a timestamp together with a message, leading to a size of $2 \cdot n \cdot G + n \cdot T + n \cdot S_m = (2 \times 65 + 4 + 20) \cdot n = 154 \cdot n$ $bytes$.
- **Zhang *et al.*** [11]: In this scheme, KGC simply generates a partial private key and a partial public key for a user. The amount of data sent by KGC in process PKG is $s + P = 20 + 40 = 60$ $bytes$ in total. In process SC, the size of data package is $(P + s + S_m) \cdot n = (40 + 20 + 20) \cdot n = 80 \cdot n$ $bytes$.
- **MCLS** [29]: In process PKG, KGC generates a partial secret key, which is an element on a pairing group, for the user, meaning that KGC only sends $G = 65$ $bytes$ to each user. In SC, to be fair and to exploit the functionality of this scheme, we assume the data package a user sends contains n messages. In this case, the total amount of a data package will be $G + 2 \cdot n \cdot s + 2 \cdot s + n \cdot S_m = 65 + 60 \cdot n + 40 = 105 + 60 \cdot n$ $bytes$.
- **RCB-BSC** [23]: In this work, in process KG, KGC generates a public key and a certificate for the user. Therefore, the amount of data sent in total is $P + s = 40 + 20 = 60$ $bytes$. As for process SC, we also assume that each data package contains n messages. Therefore, the size of the ciphertext is $(n+1) \cdot P + 2 \cdot s + n \cdot S_m = 80 + 60 \cdot n$ $bytes$.
- **Ours**: In our proposed scheme, in process PKG, KGC is supposed to send a pseudo identity along with a partial public key and a partial secret key to a user which means the communication cost is $ID + P + s + 2 \cdot T = 4 + (3 \times 4) + 40 + 20 = 72$ $bytes$. In process SC, according to Section IV, the size is $2 \cdot P + ID + T + 2 \cdot n \cdot S_m = 2 \times 40 + 2 \times 4 + 20 + 2 \cdot 20 \cdot n = 108 + 40 \cdot n$ $bytes$.

The results of comparison are shown in Fig 6 and Fig 7.

Results show that our scheme achieves a 52% cost reduction in PKG/KG compared to [31], requiring only 72 bytes versus their 150 bytes. Meanwhile, compared to [11], [23], our communication overhead is slightly heavier with an additional cost of 12 $bytes$ and 7 $bytes$ compared to [29]. This is mainly due to the pseudo identity and two timestamps generated by KGC to achieve conditional anonymity and to protect users from replay attacks. Thus, we believe our scheme offers a superior trade-off between communication efficiency and functionality.

In the scenario of SC, as illustrated in Fig 6, when there are 10 receivers, the communication cost for schemes [31], [11], [29], and [23] are 1540 $bytes$, 800 $bytes$, 705 $bytes$, and 680 $bytes$, respectively, while our scheme incurs only 508 $bytes$. This demonstrates that our scheme is significantly more efficient, with costs amounting to 32.99%, 63.5%, 72.06%, and 74.71% of the mentioned schemes. Apparently, our scheme prevails in communication efficiency.

C. Energy Consumption

In this section, we compare the energy consumption of signcrypting and unsigncrypting 100 messages using the benchmark schemes and our proposed scheme. Energy consumption is determined by the runtime of a process and the maximum power of a Raspberry Pi Model 4 B. According to the official documentation of Raspberry Pi[1], the power of a Raspberry Pi Model 4 B is $1.25A \times 5.1V = 6.375W$. Based on the runtime and the power of the devices, we can get the energy consumption during each phase, as presented in Table VI.

Compared to schemes [11], [23], [29], and [31], our scheme reduces the energy consumption of USC by 32.74%, 68.63%, 79.42%, and 78.75% respectively. For SC, the reductions are

[1]https://github.com/raspberrypi/documentation/tree/develop/documentation/asciidoc/computers/raspberry-pi

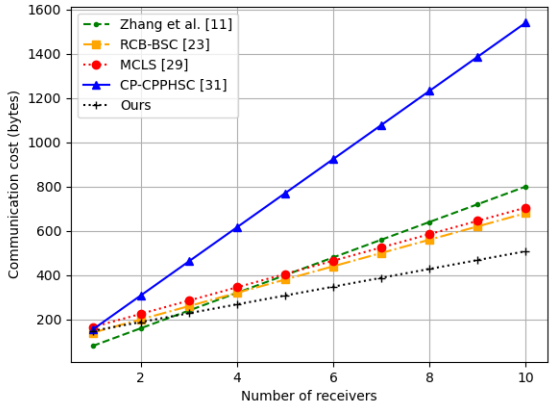

Fig. 6. Communication cost comparison of signcryption

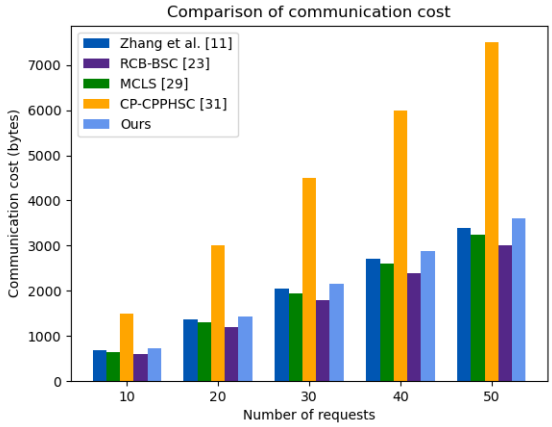

Fig. 7. Communication cost comparison of (partial) key generation

72.69%, 48.70%, 89.71%, and 89.86% respectively. These results clearly demonstrate that our scheme is highly energy-efficient and well-suited for IoV devices.

TABLE VI
COMPARISON OF ENERGY CONSUMPTION

Process / Scheme	SC	USC
[31]	31.783 J	30.188 J
[11]	11.807 J	9.540 J
[29]	31.340 J	31.172 J
[23]	6.284 J	20.451 J
Ours	3.224 J	6.416 J

D. Practicality Verification

In this section, we verify the practicality of our proposed scheme in real-world ITS by simulating a practical scenario and comparing the execution latency of our solution with the maximum allowable latency stipulated in the IEEE 802.11p standard [44].

The simulated scenario assumes a highway fully covered by a 300 Mbps 5G network, where a vehicle, as a sender, exchanges traffic data with 15 surrounding vehicles within a 250 m range, with wireless signals propagating at 80% light speed. According to IEEE 802.11p standard, the sender should exchange necessary traffic information at least once every 100 ms. To prove the practicality in our proposed scheme, this time gap should cover signcryption delay D_{SC}, propagation delay D_P, transmission delay D_T and unsigncryption delay D_{USC}. For D_{SC}, according to the previous discussion, we have $D_{SC} = 5.032 \times 15 + 2.2472 \approx 77.73\ ms$. For D_T, the length of the multi-receiver message is $108 + 40 \times 15 = 708$ $bytes$. Accordingly, $D_T = \frac{708 \times 8}{300 \times 10^6} \times 10^3 \approx 0.019\ ms$. For the farthest receiver that is 250 m away from the sender, $D_P = \frac{500}{80\% \times 3 \times 10^8} \times 10^3 \approx 0.001\ ms$. As for D_{USC}, it is always 10.06 ms as mentioned previously. To sum up, the total time consumption $D = D_{SC} + D_P + D_T + D_{USC} = 77.73 + 0.019 + 0.001 + 10.08 \approx 87.83\ ms \leq 100\ ms$. Therefore, our proposed scheme is proved to be practical in real-world ITS.

Note that these results, obtained on a Raspberry Pi 4B (4-core Cortex-A72 1.5 GHz), may appear limited compared to modern automotive processors like Tesla's 12-core FSD 3.0 (2.2 GHz). Thus, our proposed scheme can perform even better with more receivers in real-world ITS undoubtedly.

VII. Conclusion

In this paper, we propose a secure and efficient personalized multi-receiver data sharing scheme with cross-domain authentication for IoV. Our scheme leverages a novel CLSC algorithm, addressing key escrow issues and eliminating the need for CA management, thus enhancing its practicality and cost-effectiveness for real-world IoV scenarios. The absence of bilinear pairings ensures resource efficiency and low latency, critical for real-time IoV applications. Moreover, we design a pseudonym generation mechanism to achieve conditional traceability for vehicles. The ability to transmit customized messages in a single data package for multiple vehicles significantly enhances the scheme's flexibility. Formal security analysis confirms that the proposed scheme satisfies both IND-CLSC-CCA2 and EUF-CLSC-CMA security requirements, ensuring data confidentiality and unforgeablity. Furthermore, we make the implementation codes available. Experimental evaluations demonstrate that our scheme outperforms comparative schemes in computational and communication efficiency and energy consumption, proving its feasibility and practicality for real-world applications.

Acknowledgment

This work was supported in part by NSFC grant (62502446, 62125206, 62172276), in part by the Ningbo Yongjiang Talent Programme Grant 2024A-402-G, in part by the the National Key R&D Program of China under Grant 2024YFC3308304, in part by Yunnan Key Research Program grant 202402AD080004, the Shanghai Action Plan for Science, Technology and Innovation grant 24BC3201300, Startup Fund for Young Faculty at SJTU (SFYF at SJTU) grant 25X010502613, and in part by the fund of Oak Grove Ventures-School of Software Technology, Zhejiang University Blockchain Joint Lab.

REFERENCES

[1] N. Sharma and R. D. Garg, "Real-time iot-based connected vehicle infrastructure for intelligent transportation safety," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 8, pp. 8339–8347, 2023.

[2] M. Yu, "Construction of regional intelligent transportation system in smart city road network via 5g network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 2, pp. 2208–2216, 2023.

[3] F.-Y. Wang, Y. Lin, P. A. Ioannou, L. Vlacic, X. Liu, A. Eskandarian, Y. Lv, X. Na, D. Cebon, J. Ma, L. Li, and C. Olaverri-Monreal, "Transportation 5.0: The dao to safe, secure, and sustainable intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 10, pp. 10 262–10 278, 2023.

[4] F. Alanazi, "Development of smart mobility infrastructure in saudi arabia: A benchmarking approach," *Sustainability*, vol. 15, no. 4, 2023. [Online]. Available: https://www.mdpi.com/2071-1050/15/4/3158

[5] G. Cheng, Y. Wang, S. Deng, Z. Xiang, X. Yan, P. Zhao, and S. Dustdar, "A lightweight authentication-driven trusted management framework for iot collaboration," *IEEE Transactions on Services Computing*, vol. 17, no. 3, pp. 747–760, 2024.

[6] L. Deng, T. Wang, S. Feng, Y. Qu, and S. Li, "Secure identity-based designated verifier anonymous aggregate signature scheme suitable for smart grids," *IEEE Internet of Things Journal*, vol. 10, no. 1, pp. 57–65, 2023.

[7] T. Li, H. Wang, D. He, and J. Yu, "Designated-verifier aggregate signature scheme with sensitive data privacy protection for permissioned blockchain-assisted iiot," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 4640–4651, 2023.

[8] M. Barbosa and P. Farshim, "Certificateless signcryption," in *Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security*, ser. ASIACCS '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 369–372. [Online]. Available: https://doi.org/10.1145/1368310.1368364

[9] Y. Zheng, "Digital signcryption or how to achieve cost(signature & encryption) ¡¡ cost(signature) + cost(encryption)," in *Advances in Cryptology — CRYPTO '97*, B. S. Kaliski, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 165–179.

[10] H. Shao and C. Piao, "A provably secure lightweight authentication based on elliptic curve signcryption for vehicle-to-vehicle communication in vanets," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 3, pp. 3738–3747, 2024.

[11] J. Zhang, C. Dong, and Y. Liu, "Efficient pairing-free certificateless signcryption scheme for secure data transmission in iomt," *IEEE Internet of Things Journal*, vol. 11, no. 3, pp. 4348–4361, 2024.

[12] W. Li, "Multi-receiver data authorization with data search for data sharing in cloud-assisted iov," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 5, pp. 4233–4250, 2024.

[13] Y. Liang, H. Yan, and Y. Liu, "Unlinkable signcryption scheme for multi-receiver in vanets," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 9, pp. 10 138–10 154, 2023.

[14] X. Li, R. Zhu, D. Du, C. Jiang, and Z. Zhou, "Ecc-based certificateless aggregate signcryption scheme in cyber-physical power systems," *IEEE Systems Journal*, vol. 18, no. 2, pp. 893–904, 2024.

[15] Y. Wang, X. Zhang, R. Chen, H.-N. Dai, X. Wang, L. Y. Zhang, and M. Li, "Multireceiver conditional anonymous singcryption for iomt crowdsourcing," *IEEE Internet of Things Journal*, vol. 11, no. 5, pp. 8401–8413, 2024.

[16] Y. Zhou, R. Xu, Z. Qiao, B. Yang, Z. Xia, and M. Zhang, "An anonymous and efficient multimessage and multireceiver certificateless signcryption scheme for vanet," *IEEE Internet of Things Journal*, vol. 10, no. 24, pp. 22 823–22 835, 2023.

[17] F. Tong, X. Chen, C. Huang, Y. Zhang, and X. Shen, "Blockchain-assisted secure intra/inter-domain authorization and authentication for internet of things," *IEEE Internet of Things Journal*, vol. 10, no. 9, pp. 7761–7773, 2023.

[18] S. Zhang, Z. Yan, W. Liang, K.-C. Li, and B. Di Martino, "Bcae: A blockchain-based cross domain authentication scheme for edge computing," *IEEE Internet of Things Journal*, vol. 11, no. 13, pp. 24 035–24 048, 2024.

[19] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of CRYPTO 84 on Advances in Cryptology*. Berlin, Heidelberg: Springer-Verlag, 1985, p. 47–53.

[20] G. Cheng, J. Huang, Y. Wang, J. Zhao, L. Kong, S. Deng, and X. Yan, "Conditional privacy-preserving multi-domain authentication and pseudonym management for 6g-enabled iov," *IEEE Transactions on Information Forensics and Security*, pp. 1–1, 2023.

[21] M. Ramadan and S. Raza, "Secure equality test technique using identity-based signcryption for telemedicine systems," *IEEE Internet of Things Journal*, vol. 10, no. 18, pp. 16 594–16 604, 2023.

[22] M. Luo, X. Zhou, and M. Qiu, "A revocable anonymous cross-domain communication scheme for smart grid based on ring signcryption," *Peer-to-Peer Networking and Applications*, vol. 17, no. 1, pp. 125–138, 2024.

[23] Y. Gao, L. Deng, S. Feng, H. Liu, B. Li, and N. Wang, "Revocable certificate-based broadcast signcryption scheme for edge-enabled iiot," *Information Sciences*, vol. 690, p. 121540, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025524014543

[24] N. Yang, C. Tang, Q. Zhou, and D. He, "Dynamic consensus committee-based for secure data sharing with authorized multi-receiver searchable encryption," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 5186–5199, 2023.

[25] W. Yang, P. Cao, F. Zhang, and Z. Liu, "Secure pairing-free certificate-based online/offline signcryption scheme with conditional privacy preserving for vanets," *IEEE Internet of Things Journal*, vol. 12, no. 4, pp. 4435–4447, 2025.

[26] Y. Zhao, Y. Wang, Y. Liang, H. Yu, and Y. Ren, "Identity-based broadcast signcryption scheme for vehicular platoon communication," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 6, pp. 7814–7824, 2023.

[27] X. D. Yang, W. J. Wang, B. Shu, M. J. Li, R. X. Liu, and C. F. Wang, "Multi-message multi-receiver signcryption scheme based on blockchain," *Mathematical Biosciences and Engineering*, vol. 20, no. 10, pp. 18 146–18 172, 2023.

[28] G. Xu, X. Yin, and X. Li, "Lightweight and secure multi-message multi-receiver certificateless signcryption scheme for the internet of vehicles," *Electronics*, vol. 12, no. 24, 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/24/4908

[29] A. Umrani, A. K. Vangujar, and P. Palmieri, "A multi-receiver certificateless signcryption (mcls) scheme," in *2024 8th International Conference on Cryptography, Security and Privacy (CSP)*, 2024, pp. 46–52.

[30] Y. Chen, J. Zhang, X. Wei, Y. Wang, and J. Cui, "Cross-domain authentication scheme for vehicles based on given virtual identities," *IEEE Internet of Things Journal*, vol. 11, no. 9, pp. 15 869–15 879, 2024.

[31] I. Ali, Y. Chen, N. Ullah, M. Afzal, and W. HE, "Bilinear pairing-based hybrid signcryption for secure heterogeneous vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 5974–5989, 2021.

[32] M. Seifelnasr, R. AlTawy, and A. Youssef, "A conditional privacy-preserving protocol for cross-domain communications in vanet," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 4, pp. 5251–5263, 2025.

[33] S. Xu, X. Chen, Y. Guo, S.-M. Yiu, S. Gao, and B. Xiao, "Efficient and secure post-quantum certificateless signcryption with linkability for iomt," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 1119–1134, 2025.

[34] J. Cui, Y. Zhu, H. Zhong, Q. Zhang, C. Gu, and D. He, "Efficient blockchain-based mutual authentication and session key agreement for cross-domain iiot," *IEEE Internet of Things Journal*, vol. 11, no. 9, pp. 16 325–16 338, 2024.

[35] C. Wang, Y. Zhang, Q. Zhang, X. Xu, W. Chen, and H. Li, "Secas: Secure and efficient cross-domain authentication scheme based on blockchain for space ttc networks," *IEEE Internet of Things Journal*, vol. 11, no. 16, pp. 26 806–26 818, 2024.

[36] Z. Wang, Z. Zong, F. Li, S. Sun, and P. Zhao, "Revocable certificateless cross-domain authentication scheme based on primary–secondary blockchain," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 5, pp. 5880–5891, 2024.

[37] X. Hao, W. Ren, Y. Fei, T. Zhu, and K.-K. R. Choo, "A blockchain-based cross-domain and autonomous access control scheme for internet of things," *IEEE Transactions on Services Computing*, vol. 16, no. 2, pp. 773–786, 2023.

[38] L. Feng, F. Qiu, K. Hu, B. Yu, J. Lin, and S. Yao, "Cabc: A cross-domain authentication method combining blockchain with certificateless signature for iiot," *Future Generation Computer Systems*, vol. 158, pp. 516–529, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X24001717

[39] Q. Miao, T. Ren, J. Dong, Y. Chen, and W. Xu, "A 3c authentication: A cross-domain, certificateless, and consortium-blockchain-based authentication method for vehicle-to-grid networks in a smart grid," *Symmetry*, vol. 16, no. 3, 2024. [Online]. Available: https://www.mdpi.com/2073-8994/16/3/336

[40] J. Dong, G. Xu, C. Ma, J. Liu, and U. G. O. Cliff, "Blockchain-based certificate-free cross-domain authentication mechanism for industrial

internet," *IEEE Internet of Things Journal*, vol. 11, no. 2, pp. 3316–3330, 2024.
[41] Q. Zhan, M. Luo, and M. Qiu, "An efficient multimode certificateless ring signcryption scheme in vanets," *IEEE Internet of Things Journal*, vol. 11, no. 20, pp. 33 508–33 524, 2024.
[42] L. Benarous, S. Zeadally, S. Boudjit, and A. Mellouk, "A review of pseudonym change strategies for location privacy preservation schemes in vehicular networks," *ACM Comput. Surv.*, vol. 57, no. 8, Mar. 2025. [Online]. Available: https://doi.org/10.1145/3718736
[43] W. Yang, P. Cao, and F. Zhang, "A secure pairing-free certificateless online/offline signcryption scheme with batch verification for edge computing-based vanets," *IEEE Transactions on Vehicular Technology*, vol. 74, no. 1, pp. 1570–1583, 2025.
[44] S. Eichler, "Performance evaluation of the ieee 802.11p wave communication standard," in *2007 IEEE 66th Vehicular Technology Conference*, 2007, pp. 2199–2203.

Taolong Su is currently a graduate student in the School of Software Techonology, Zhejiang University, Hangzhou, China. He received a bachelor's degree in the School of Civil Engineering, Sun Yat-sen University, Guangdong, China, in 2022. His research interests lie in the fields of data security, data fusion, and Internet-of-Things.

Guanjie Cheng is currently an Assistant Professor in the School of Software Techonology, Zhejiang University, Hangzhou, China. He received a Ph.D. degree in College of Computer Science and Technology, Zhejiang University, Hangzhou, China, in 2023, and a bachelor's degree in the School of Computer Science and Technology, Xidian University, Xian, China, in 2018. He was a visiting Ph.D. in Nanyang Technological University, Singapore, in 2023. He is a long-term reviewer for the top journal Chinese Journal of Electronics. His research interests lie in the fields of data security, edge intelligence, Internet-of-Things, and blockchain.

Junqin Huang is currently a research assistant professor at the School of Cyber Science and Engineering, Shanghai Jiao Tong University. He received the Ph.D. Degree in computer technology from Shanghai Jiao Tong University in 2024 and the B.Eng. degree in computer science and technology from University of Electronic Science and Technology of China in 2018. His research interests include blockchain, internet of things, data security, trusted computing.

Xinkui Zhao is a ZJU 100-Young Professor at Zhejiang University, China. His research interests primarily focus on Cloud-Native Architecture and Technologies, Intelligent Operating Systems, and Service Computing. He has led the development of multiple large-scale enterprise-level cloud-native platforms and authored over 30 academic papers published in top-tier journals and conferences, including ASPLOS, DAC, WWW, SCIS, TPDS, TSC, ICWS, ICSOC, and more.

Shuiguang Deng (Senior Member, IEEE) is currently a full professor at the College of Computer Science and Technology in Zhejiang University, China, where he received a BS and PhD degree both in Computer Science in 2002 and 2007, respectively. He previously worked at the Massachusetts Institute of Technology in 2014 and Stanford University in 2015 as a visiting scholar. His research interests include Edge Computing, Service Computing, and Blockchain. He serves for the journal IEEE Trans. on Services Computing, Knowledge and Information Systems, Computing, and IET Cyber-Physical Systems: Theory & Applications as an Associate Editor. Up to now, he has published more than 100 papers in journals and refereed conferences.