electronics

MDPI

Article

Lightweight and Secure Multi-Message Multi-Receiver Certificateless Signcryption Scheme for the Internet of Vehicles

Guishuang Xu [1,2], Xinchun Yin [1,2,3,*] and Xincheng Li [1,2]

[1] College of Information Engineering, Yangzhou University, Yangzhou 225127, China; mx120210552@yzu.edu.cn (G.X.); 18752782261@163.com (X.L.)
[2] Henan Key Laboratory of Network Cryptography Technology, Information Engineering University, Zhengzhou 450001, China
[3] Guangling College, Yangzhou University, Yangzhou 225009, China
* Correspondence: xcyin@yzu.edu.cn

Abstract: The Internet of Vehicles (IoV) improves traffic efficiency and enhances driving safety through the real-time collection and analysis of traffic-related data. Numerous secure and privacy-preserving communication protocols have been proposed for the IoV. However, various security threats, privacy leakage, and inefficient communications remain unaddressed. Therefore, a lightweight and secure multi-message multi-receiver certificateless signcryption (LS-MRCLSC) scheme based on elliptic curve cryptography (ECC) is proposed. The proposed scheme guarantees secure communication and promotes messaging efficiency with multi-cast mode. Multiple key generation centers (KGCs) collaborate to generate and update the system master key (SMK) using Feldman's verifiable secret-sharing (FVSS) algorithm, avoiding the single point of failure (SPoF) problem. Formal security proofs under the random oracle model (ROM) demonstrate that the proposed scheme meets requirements such as data confidentiality, message unforgeability, anonymity, and unlinkability. Performance evaluations confirm that the LS-MRCLSC scheme is better than similar schemes in terms of efficiency, feasibility, and scalability.

Keywords: internet of vehicles; IoV; certificateless signcryption; multi-message multi-receiver; resisting KGC damage attacks

Citation: Xu, G.; Yin, X.; Li, X. Lightweight and Secure Multi-Message Multi-Receiver Certificateless Signcryption Scheme for the Internet of Vehicles. *Electronics* **2023**, *12*, 4908. https://doi.org/10.3390/electronics12244908

Academic Editors: Muath Obaidat and Kutub Thakur

Received: 30 October 2023
Revised: 2 December 2023
Accepted: 4 December 2023
Published: 6 December 2023

1. Introduction

With the rising number of social vehicles, traffic accidents are becoming more frequent, and urban areas are experiencing traffic congestion, which poses a significant barrier to the economic growth of cities. The Internet of Vehicles (IoV) integrates cutting-edge vehicle sensors, controllers, actuators, and modern communication technology to enable intelligent information sharing and interaction between vehicles, people, roads, and the cloud. Vehicles can transmit real-time traffic information (e.g., current location, speed, weather conditions, and road congestion) among IoV infrastructures through dedicated short-range communications (DSRC) or cellular vehicle-to-everything (C-V2X) [1] standards. This helps other vehicles plan more efficient traffic routes and reduces the occurrence of traffic accidents. However, numerous security threats exist during data transmission due to insecure open wireless channels. Attackers may eavesdrop, forge, delete, replay, and tamper with the transmitted data. Certificateless signcryption [2] is a solution for ensuring communication security by providing both message authentication and data confidentiality simultaneously. Nevertheless, in certificateless signcryption protocols, receivers decrypt the ciphertext to access the message and then verify the validity of the signature, which leads to massive computation delays. Absolute anonymity is undesirable because of the requirement of supervision. A traffic management authority (TMA) should have the capability to trace and recover the real identity of malicious vehicles that have sent fake or faulty messages to

disturb traffic orders [3]. Therefore, many schemes adopt pseudonym technology to protect the privacy of vehicles (e.g., identity, location, request content).

Certificateless communication protocols leverage a key generation center (KGC) to generate partial private keys (PPKs) and eliminate key escrow problems. Nevertheless, a KGC is not totally credible and is susceptible to denial of service (DoS) attacks in the IoV. With the success of advanced persistent threat (APT) attacks, attackers can acquire the system master key (SMK) and seriously threaten the security of the system. To resolve this issue, researchers have proposed the utilization of multiple KGCs to manage an SMK with Shamir's threshold secret-sharing (TSS) scheme [4]. Consequently, attackers have to corrupt at least threshold KGCs to retrieve the SMK. However, this cannot guarantee the safety of KGCs since it is possible for threshold KGCs to be corrupted in practical scenarios. Furthermore, Shamir's TSS scheme involves a key distributor that knows the secret key, and the holder of the sub-key may provide an unreal share. Thus, devising a dynamically updatable protocol that allows KGCs to update the SMK for signcryption will enhance the security of the system. Even if attackers recover the SMK of the last period, they cannot disturb the current status and operation of the system.

Furthermore, the computation and storage capacities of the on-board unit (OBU) loaded onto vehicles are significantly limited. In the IoV, vehicles are equipped with multiple sensors that can detect various heterogeneous messages simultaneously, including informative messages (speed, weather conditions), indicative messages (direction, coordinates), and emergency messages (traffic accidents, traffic jams, natural disasters) [5]. For different types of messages, we aim to send them to the corresponding receivers as quickly as possible. For instance, if vehicles need to send emergency messages to a TMA to optimize road conditions, the best course of action is to transmit these messages to the nearest roadside unit (RSU). An RSU is capable of verifying the received messages and broadcasting them to inform the TMA, nearby wired-connected RSUs, and vehicles within its communication range. This allows the TMA to be aware of the current traffic conditions and take real-time management measures while also ensuring that the RSU spreads messages over the maximum range as fast as possible. Regarding indicative messages, direct vehicle-to-vehicle (V2V) communication will aid receivers in facilitating traffic strategies. In traditional signcryption schemes, sender vehicles must execute the signcryption algorithm n times to send n messages, which is a significant challenge for OBUs with limited resources in a delay-sensitive IoV. Therefore, reducing the overall computation costs required for the signcryption algorithm is crucial.

All in all, there are various challenges in the IoV: (a) insecure communication; (b) privacy leakage of vehicles; (c) key escrow problems; (d) single points of failure (SPoFs) of KGCs; and (e) inefficient message transmission. Thus, this research aims to design a secure and lightweight communication scheme for the IoV to address the aforementioned challenges.

1.1. Contribution

Regarding the above concerns, a lightweight and secure multi-message multi-receiver certificateless signcryption (LS-MRCLSC) scheme with multiple KGCs for the IoV is proposed, based on a multi-message multi-receiver signcryption (MMSC) scheme [2]. Our main contributions can be summarized as follows:

1. **An LS-MRCLSC scheme with multiple KGCs is proposed.** The proposed LS-MRCLSC scheme is proven to realize confidentiality and unforgeability under the intractable problems in the random oracle model (ROM). Furthermore, it achieves the fundamental security requirements of the IoV such as anonymity, unlinkability, forward and backward secrecy, and resistance to KGC damage attacks and replay attacks.
2. **Multiple KGCs are employed in the LS-MRCLSC instead of the traditional single KGC, which avoids SPoFs and key escrow problems.** With Feldman's verifiable secret-sharing (FVSS) [6] mechanism, multiple KGCs negotiate the SMK after a round of communication. To resist APT attacks, each KGC is able to periodically update its

own sub-key and the SMK. Moreover, secure channels are not required during PPK transmission, which improves the robustness of our LS-MRCLSC scheme.

3. **The LS-MRCLSC scheme effectively reduces the computation cost and communication overhead.** Both theoretical analysis and simulation experiments demonstrate that the LS-MRCLSC scheme is efficient in terms of computation cost and communication overhead. Specifically, when there are 100 receivers, the total computation time (signcryption and unsigncryption) of the LS-MRCLSC scheme is reduced by 48.77%, 66.28%, 48.90%, 49.27%, and 49.27%, respectively, compared to the schemes in [7–11]. In addition, the communication overhead is reduced by 7.32%, 83.57%, 47.06%, 0.93%, and 0.93%, respectively, compared to the schemes in [7–11].

1.2. Organization

The remainder of this article is organized as follows. Section 2 outlines the related works. In Section 3, the preliminaries, including complexity assumption, system model, security model, and security goals, are introduced. The proposed LS-MRCLSC scheme is presented in Section 4. Section 5 demonstrates the security proof and analysis. The performance evaluation results are given in Section 6. Finally, we conclude this work in Section 7.

2. Related Works

2.1. Conditional Privacy-Preserving Schemes in the IoV

To ensure the secure communication of vehicles, a series of protocols have been proposed with cryptographic technology. Cui et al. [12] devised an efficient authentication scheme based on semi-trusted authority, which combines self-repairing key distribution and certificate signing. But it increases the communication overhead of the system. Gao et al. [13] introduced a decentralized distributed denial of service (DDoS) attack detection scheme using big data techniques. The scheme mainly comprises two parts: real-time network traffic acquisition and network traffic detection. Nevertheless, the authors did not perform simulated attacks to analyze the performance of the system. Baza et al. [14] proposed a scheme to detect Sybil attacks using proof-of-work (PoW) and proof-of-trajectory (PoT) mechanisms, combining both trajectory verification and resource testing. However, the method could fail if a capable attacker focused on endowing a fake vehicle with additional computational resources, causing confusion in a specific region. In 2022, to resist attacks like eavesdropping, Ren et al. [15] proposed an efficient distance-based privacy-preserving authentication protocol. They used hash functions and exclusive-OR operations to fulfill the privacy protection requirement, which is based on distance. However, it cannot effectively withstand malicious tampering attacks. To cope with challenges like the leakage of data and personal privacy, Bao et al. [16] introduced a scheme with dynamic service, which attains full policy hiding by implementing access control in the inner product. Moreover, they designed an efficient indirect revocation mechanism, which enables the cloud and users to update the ciphertext and user secret key. Recently, blockchain technology has become popular due to its tamper-proof nature, decentralization, and transparency. Conditional privacy-preserving authentication (CPPA) protocols based on blockchain [17,18] have been devised. They utilize blockchain technology for storing vehicular certificates to realize effective certificate management. Tu et al. [19] proposed a vehicle-based secure blockchain consensus algorithm, which overcomes the leakage of sensitive data, high costs, and delays. In addition, homomorphic techniques have gradually been applied to the CPPA protocols for the IoV. Verma et al. [20] utilized homomorphic signatures to protect the confidentiality and unforgeability of traffic-related messages. Homomorphic cryptography [21] assists in accomplishing tripe pseudonym authentication, reducing the dependence on TMAs.

2.2. Certificateless Signcryption Schemes

In 2003, Al-Riyami and Paterson [22] first proposed the concept of certificateless public key cryptography, which addressed the certificate management problem in traditional

public key infrastructure (PKI) schemes [23] and the key escrow problem in ID-based signature schemes [24]. Since then, scholars have proposed many certificateless (aggregate) signature schemes [25–28]. However, these schemes cannot resist public key replacement attacks or malicious KGC attacks. Moreover, in the signature system, sender vehicles send out traffic messages along with the signature. Although attackers cannot steal secret keys or forge signatures via eavesdropping wireless channels, they can know the message content from intercepted data, which may contain sensitive information. Therefore, it is vital to ensure data confidentiality. The signcryption primitive was proposed by Zheng et al. [29], which performs signature and encryption in one logical step. Nevertheless, if we want to send a secret message to multiple receivers, the traditional one-to-one structure of signcryption would no longer be efficient. Selvi et al. [30] first proposed the concept of multi-receiver certificateless signcryption (MRCLSC) and provided a security model. However, Miao et al. [31] pointed out that it cannot maintain confidentiality under internal attacks. Receiver anonymity is also significant in the IoV, which means that each user can identify whether they are an authenticated receiver but cannot identify others. Focusing on the privacy issue of heterogeneous systems, Niu et al. [32] constructed an aggregate signcryption scheme based on MRCLSC. Li and Pang [33] declared that Niu et al.'s scheme [32] could not really achieve receiver anonymity because of the fixed Lagrange interpolation polynomial results. Then, Pang et al. [34] devised an efficient MRCLSC scheme without bilinear pairing, while Yu et al. [35] substantiated that Pang's et al. scheme [34] could not achieve unforgeability and confidentiality, as the adversary can randomly forge the public and private key pairs of users. Yu et al. [35] also found that the schemes in [36–38] could not ensure the integrity of transmitted data. Considering the secure data transmission of wireless body area networks, Shen et al. [7] proposed a lightweight MRCLSC scheme. However, they utilized secure channels to transmit PPKs.

Moreover, the existing multi-receiver signcryption (encryption) schemes cannot send multiple different messages to multiple related vehicles in a data report. Seo and Kim [39] proposed the first MMSC scheme, which supports sending n messages at a time. Soon after, MMSC schemes based on chaotic theory [40] and elliptic curve cryptography (ECC) [41] were proposed. However, receivers can obtain all plaintext by decrypting one ciphertext in these schemes. Zhou et al. [8] presented a certificateless MMSC scheme to realize the anonymous transmission of multiple messages in multicast communication, but the overhead is extremely high due to bilinear pairing. Pang et al. [42] presented a certificateless MMSC scheme, claiming it was secure and efficient. However, Peng et al. [9] proved that Pang et al.'s scheme [42] is vulnerable to a Type I attack. Therefore, the confidentiality, unforgeability, and anonymity of the senders cannot be guaranteed. Although some MMSC schemes have been designed [43–45], Pang et al. [42] pointed out that none of them can provide receiver anonymity and privacy preservation. Qiu et al. [46] proposed an MMSC scheme for a heterogeneous smart mobile Internet of Things (IoT), but it cannot achieve receiver anonymity. Recently, Ming et al. [10] devised an MMSC scheme for the healthcare IoT. Nevertheless, it is unable to withstand replay attacks. Zhou et al. [11] also proposed an anonymous certificateless MMSC scheme for a vehicular ad hoc network (VANET). However, the scheme cannot resist replay attacks, and unlinkability is not achieved.

Secure channels are required when a KGC generates PPKs for users in the above schemes. However, maintaining secure channels needs huge economic expenditure. In addition, most existing schemes utilize a single KGC. If attackers successfully invade a KGC, the security of the system will be seriously threatened. Moreover, certificateless signature schemes cannot avoid the key escrow problem. Hence, it is important to guarantee the security of the KGC in the LS-MRCLSC scheme.

3. Preliminaries

The preliminaries of the LS-MRCLSC scheme are introduced in this section.

3.1. Notations

The main notations and corresponding descriptions of our scheme are presented in Table 1. The abbreviations used in this paper are listed in the abbreviation table following Section 7.

Table 1. Notations and corresponding descriptions.

Notation	Description
q	A large prime number
$\mathbb{G}$	An addictive cyclic group with order q
H_i	Secure one-way hash function
P	A generator of $\mathbb{G}$
$params$	System's public parameters
V_i	The i-th vehicle
V_{R_i}	The i-th receiver vehicle
KGC_i	The i-th KGC
t	Threshold value
s_i	Sub-key of KGC_i
P_i	Sub-public key of KGC_i
s	The SMK
P_{pub}	System public key
a	Private key of TMA
T_{pub}	Public key of the TMA
F_{index}	An index function
$\oplus$	XOR operation
RID_i	The real identity of V_i
ID_i	The temporary identity of V_i
$PID_{i,j}$	The j-th pseudo-identity of V_i
$T_{i,j}$	Valid period of $PID_{i,j}$
k_i	The temporary PPK of V_i
d_i	The PPK of V_i
x_i	The secret key of V_i
(PK_i, SK_i)	Public and private key pair of V_i
m_{R_i}	Traffic message related to V_{R_i}
M	Traffic message set to be signcrypted
sig_{R_i}	Signature related to V_{R_i}
c_{R_i}	Ciphertext related to V_{R_i}
$t_{R_i,j}$	The j-th current timestamp related to V_{R_i}
CT	Ciphertext set
T	Timestamp set
C_m	Signcryption ciphertext
A	The adversary
C	The challenger
ε	The probability that adversary A wins the game
Δtt	The valid time interval

3.2. Complexity Assumption

In this subsection, the complexity assumptions associated with the ECCDHP and ECDLP are introduced.

Elliptic Curve Computational Diffie–Hellman Problem (ECCDHP): Given an elliptic curve E, choose a group $\mathbb{G}$ on E, where $\mathbb{G}$ has the prime order q and generator P. Given a tuple $(aP, bP) \in \mathbb{G}$, it is difficult to compute abP in probabilistic polynomial time (PPT), where $a, b \in \mathbb{Z}_q^*$.

Elliptic Curve Discrete Logarithm Problem (ECDLP): Given an elliptic curve E, choose a group $\mathbb{G}$ on E, where $\mathbb{G}$ has the prime order q and generator P. Given a tuple $(P, W) \in \mathbb{G}$, it is difficult to compute $a \in \mathbb{Z}_q^*$ in PPT, where $W = aP$.

3.3. Feldman's Verifiable Secret Sharing

Let $\mathbb{G}$ be a group with order q. The sharing algorithm takes the threshold parameters LL and t and a secret $ss \in \mathbb{Z}_q^*$; chooses a polynomial with random coefficients, except for the constant term, i.e., $p(X) = a_0 + a_1X + \ldots + a_tX^t (a_0 = ss)$; and outputs the commitments $A_k = g^{a_k} \in \mathbb{G}$ for $k = 0, 1, \ldots, t$. The j-th share ss_j is $p(j)$ for $j = 1, \ldots, LL$.

To verify the j-th share against the commitments, the verification algorithm takes ss_j and a set of commitments $\{A_k\}_{k=0}^{t}$ and checks whether $g^{ss_j} = \prod_{k=0}^{t} (A_k)^{j^k}$. The above algorithms are defined as:

- $FShare(ss, t, LL) \rightarrow \{ss_j\}_{j=1}^{LL}, \{A_k\}_{k=0}^{t}$.
- $FVerify(ss_j, \{A_k\}_{k=0}^{t}) \rightarrow b$, where $b \in \{0, 1\}$.

3.4. System Model

Figure 1 describes the system model of our LS-MRCLSC scheme. There are four entities equipped in vehicles: the TMA, KGC, RSUs, and OBUs. The upper layer is composed of the TMA and KGC, which communicate over the wired channels. The lower layer consists of RSUs and vehicles, which communicate over the wireless channels.

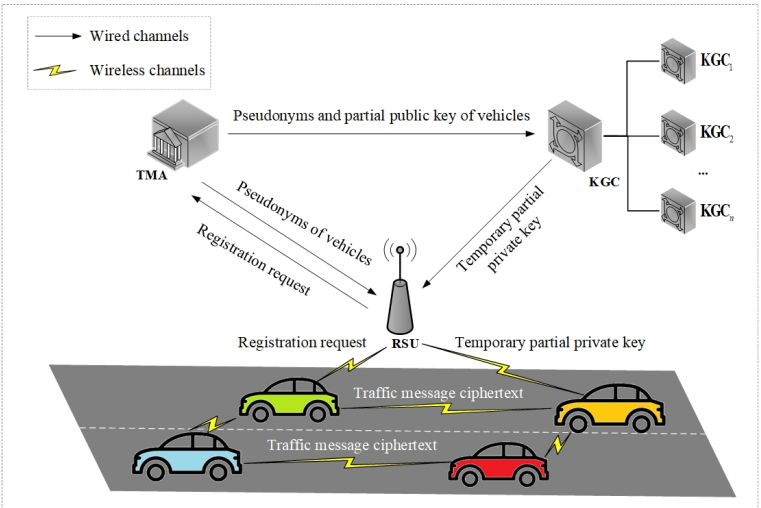

Figure 1. The system model of our LS-MRCLSC scheme.

1. TMA: The TMA is usually a trusted traffic management department that is responsible for generating the system's public parameters. Moreover, the TMA helps vehicles generate pseudonyms and traces the real identities of malicious vehicles if necessary.
2. KGC: The KGC, composed of $KGC_i (i = 1, 2, \ldots, n)$, is responsible for generating the PPKs for vehicles through public channels. KGC_i cooperatively negotiates the SMK with $KGC_j (j = 1, 2, \ldots, n, j \neq i)$. It could be compromised if attackers achieve a specific threshold t within one epoch. Hence, $KGC_i (i = 1, 2, \ldots, n)$ should periodically update their own sub-keys and SMKs.
3. RSU: The RSU is the communication equipment installed along the roadside. RSUs can receive and verify the traffic messages from vehicles within their communication range. After verifying the validity of messages, they provide services like network connections for vehicles.
4. Vehicles: Each vehicle is equipped with an OBU to sense, compute, and process traffic data. OBUs signcrypt multiple traffic messages and send ciphertext to RSUs or other vehicles. Meanwhile, vehicles can be the receivers, decrypt ciphertext, and obtain traffic data.

3.5. Security Model

According to the definition in [22], a signature system in the IoV faces two types of attacks. A Type I attacker is malicious, denoted as V_i. It can attack the security of the LS-MRCLSC scheme and replace the public keys of all vehicles. A Type II attack is a malicious KGC. It can obtain the SMK but cannot replace a user's public key. Meanwhile, the LS-MRCLSC scheme should simultaneously offer indistinguishability against a chosen ciphertext attack adaptively (IND-CCA2) and existential unforgeability under a chosen message attack (EUF-CMA).

Game I: This game is played between challengers $C_{\text{I-1}}$ and $A_{\text{I-1}}$.

Definition 1. *Confidentiality against a Type I attack. If there is no adversary, $A_{I\text{-}1}$ can win Game I with a non-negligible probability in polynomial time, so the LS-MRCLSC scheme offers IND-CCA2.*

1. $C_{\text{I-1}}$ executes the system initialization algorithm, generates the system parameters $params$, and returns the results to $A_{\text{I-1}}$.
2. $C_{\text{I-1}}$ executes the following queries adaptively.
   (1) PPK generation query: $A_{\text{I-1}}$ chooses an identity $PID_{i,j}$, and challenger $C_{\text{I-1}}$ computes $(R_i, d_i) \leftarrow PPKgeneration(params, s, PID_{i,j})$ and returns it to $A_{\text{I-1}}$.
   (2) Private key generation query: $A_{\text{I-1}}$ chooses an identity $PID_{i,j}$, and challenger $C_{\text{I-1}}$ computes $d_i \leftarrow \text{PPKGen}(params, s, PID_{i,j})$, $x_i \leftarrow \text{SecretKeyValue}(params, PID_{i,j})$, and $SK_i \leftarrow \text{PrivateKeyGen}(params, PID_{i,j}, x_i, d_i)$ and returns SK_i to $A_{\text{I-1}}$.
   (3) Public key generation query: $A_{\text{I-1}}$ chooses an identity $PID_{i,j}$, and challenger $C_{\text{I-1}}$ computes $R_i \leftarrow \text{PPKGen}(params, s, PID_{i,j})$, $X_i \leftarrow \text{SecretKeyValue}(params, PID_{i,j})$, and $PK_i \leftarrow \text{PublicKeyGen}(params, PID_{i,j}, X_i, R_i)$ and returns PK_i to $A_{\text{I-1}}$.
   (4) Signcryption query: $A_{\text{I-1}}$ chooses PID_a, PID_b, and message m, and challenger $C_{\text{I-1}}$ executes a private key generation query for PID_a and public key generation queries for PID_a and PID_b, computes $C_m = \text{Signcryption}(m, SK_a, PK_a, PK_b)$, and returns C_m to $A_{\text{I-1}}$.
   (5) UnSigncryption query: $A_{\text{I-1}}$ chooses PID_a, PID_b, and ciphertext C_m, and challenger $C_{\text{I-1}}$ executes a private key generation query for PID_b and public key generation queries for PID_a and PID_b, computes $\text{UnSigncryption}(C_m, SK_b, PK_a, PK_b)$, and returns m to $A_{\text{I-1}}$.
   (6) Public key replacement: At any time, $A_{\text{I-1}}$ chooses a new value to replace PK_i.
3. $A_{\text{I-1}}$ generates two isometric messages m_0, m_1 and two identities PID_a, PID_b as a challenge, where PID_b cannot be the identity that has executed the PPK generation query or private key generation query. $C_{\text{I-1}}$ randomly chooses $j \in \{0, 1\}$, computes $C_m = \text{Signcryption}(m_j, SK_a, PK_a, PK_b)$, and returns C_m to $A_{\text{I-1}}$.
4. In the guess stage, $A_{\text{I-1}}$ executes polynomial bounded degree queries similar to step 2, but it cannot execute the PPK generation query, private key generation query, or UnSigncryption query for C_m.
5. $A_{\text{I-1}}$ outputs j' as the guess of j. If $j' = j$, $A_{\text{I-1}}$ wins Game I.

Game II: This game is played between challengers $C_{\text{II-1}}$ and $A_{\text{II-1}}$.

Definition 2. *Confidentiality against a Type II attack. If there is no adversary, $A_{II\text{-}1}$ can win Game II with a non-negligible probability in polynomial time, so the LS-MRCLSC scheme offers IND-CCA2.*

1. $C_{\text{II-1}}$ executes the system initialization algorithm, generates the system parameters $params$, and returns $params$ and s to $A_{\text{II-1}}$.
2. $C_{\text{II-1}}$ executes queries adaptively, as in Definition 1, except for the PPK generation query and public key replacement.
3. The challenge stage and guess stage are the same as in Definition 1.
4. $A_{\text{II-1}}$ outputs j' as the guess of j. If $j' = j$, $A_{\text{II-1}}$ wins Game II.

Game III: This game is played between challengers $C_{\text{I-2}}$ and $A_{\text{I-2}}$.

Definition 3. *Unforgeability against a Type I attack. If there is no adversary, $A_{I\text{-}2}$ can win Game III with a non-negligible probability in polynomial time, so the LS-MRCLSC scheme offers EUF-CMA.*

1. $A_{\text{I-2}}$ executes steps 1 and 2, as in Definition 1.
2. $A_{\text{I-2}}$ outputs a new signature $\{C_m^{'}, PID_{i,j}\}$. Moreover, $PID_{i,j}$ does not execute a PPK generation query, private key generation query, or signcryption query. If the result of UnSigncryption $(C_m^{'}, PID_{i,j})$ is '1', then $A_{\text{I-2}}$ wins Game III.

Game IV: This game is played between challengers $C_{\text{II-2}}$ and $A_{\text{II-2}}$.

Definition 4. *Unforgeability against a Type II attack. If there is no adversary, $A_{II\text{-}2}$ can win Game IV with a non-negligible probability in polynomial time, so the LS-MRCLSC scheme offers EUF-CMA.*

1. $A_{\text{II-2}}$ executes steps 1 and 2, as in Definition 2.
2. $A_{\text{II-2}}$ outputs a new signature $\{C_m^{'}, PID_{i,j}\}$. Moreover, $PID_{i,j}$ does not execute a PPK generation query, private key generation query, or signcryption query. If the result of UnSigncryption $(C_m^{'}, PID_{i,j})$ is '1', then $A_{\text{II-2}}$ wins Game IV.

3.6. Security Goals

The LS-MRCLSC scheme should fulfill the following security requirements:

1. Data confidentiality: The traffic data should be encrypted during transmission, and only the designated receivers can decrypt the corresponding ciphertext.
2. Message unforgeability: The LS-MRCLSC scheme can resist signature forgeability attacks. Receivers (V_i or RSUs) can verify the validity of signatures to confirm that the messages were sent by valid vehicles and not tampered with during transmission.
3. Anonymity: Vehicles should utilize pseudonyms to communicate with others. Apart from the TMA, any third-party entity cannot know the real identities of registered vehicles.
4. Unlinkability: No vehicle, RSU, or other third party can judge whether two or more messages are from the same vehicle. In other words, attackers cannot trace vehicles through messages over public channels.
5. Resist KGC damage attacks: Attackers cannot steal the system master key when they compromised fewer than t KGC_i ($t < n$, where n denotes the total number of KGC_i) in the same period. Even if the current SMK is disclosed, attackers cannot obtain the previous or subsequent communication keys.
6. Forward and backward secrecy: Although the private keys of vehicles and KGCs are disclosed in the current period, attackers cannot obtain the previous or subsequent private keys.
7. Resistance to replay attack: This prevents attackers from re-transmitting messages that were eavesdropped over public channels.

4. The LS-MRCLSC Scheme for the IoV

4.1. High-Level Description

Our LS-MRCLSC scheme contains six stages: system initialization, pseudonym generation, key generation, message signcryption and unsigncryption, KGC secret key update, and malicious vehicle tracing. Figure 2 shows the detailed process.

In the system initialization stage, the TMA and KGC generate their own private keys and system public parameters. In the pseudonym generation stage, vehicle registration is accomplished through the TMA. In the key generation stage, vehicles compute complete public and private keys with the help of KGC_i. In the message signcryption and unsigncryption stage, the sender vehicle signcrypts the messages and sends the ciphertext to other

vehicles or RSUs. Then, the receiver vehicles or RSUs unsigncrypt the ciphertext to obtain the traffic-related information. In the KGC secret key update stage, $KGC_i (i = 1, 2, \ldots, n)$ update their own sub-keys and SMKs by executing the FVSS algorithm. In the malicious vehicle tracing stage, the TMA retrieves the real identities of malicious vehicles and punishes them.

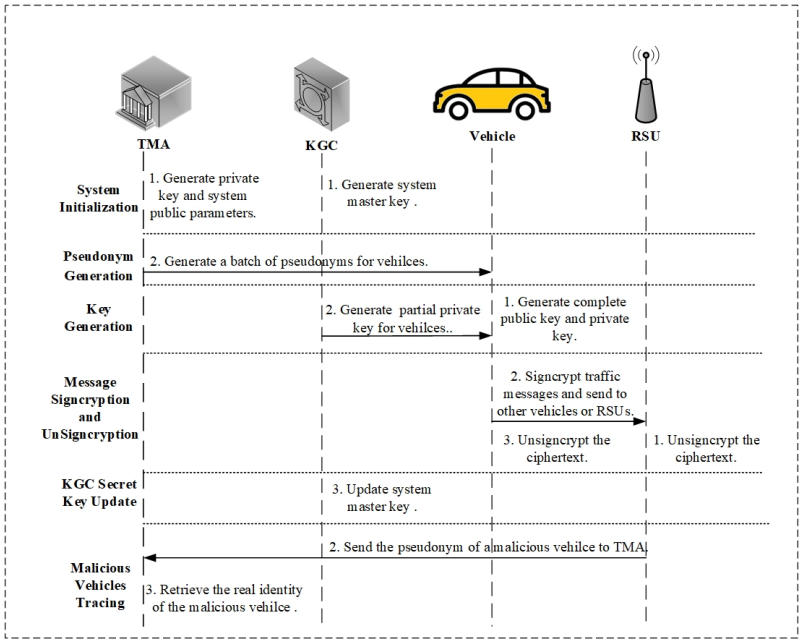

Figure 2. Overview of the LS-MRCLSC scheme.

4.2. System Initialization

In this stage, the TMA and KGC generate the public system parameters, which is the public input in the later stages.

1. **TMA initialization.** The TMA generates its private key, which is used to register vehicles. Furthermore, the TMA generates some of the public system parameters. The steps are as follows:
   (1) Input security parameter λ, and let $\mathbb{G}$ be an additive cyclic group generated by P with prime order q.
   (2) Randomly choose $a \in \mathbb{Z}_q^*$ as its private key, store it secretly, and set the public key $T_{pub} = aP$.
   (3) To fulfill the security requirements of the proposed scheme, six secure one-way hash functions are selected. $H_0 : \mathbb{G} \to \mathbb{Z}_q^*$ is used to achieve key agreement [47] in pseudonym generation requests and PPK generation, which enables the PPK to be transmitted over public channels. $H_1 : \mathbb{G} \times \mathbb{Z}_q^* \times \mathbb{Z}_q^* \to \mathbb{Z}_q^*$ is used to request pseudonyms, which enables the registration request to be transmitted over public channels. $H_2 : \mathbb{G} \times \{0,1\}^* \to \mathbb{Z}_q^*$ is used to generate pseudonyms for vehicles, which provides a secure link between the real identity and pseudo-identity of vehicles to the TMA. If the vehicle sends fake or faulty messages, the TMA can retrieve its real identity from the pseudonym. $H_3 : \mathbb{Z}_q^* \times \mathbb{G} \times \mathbb{G} \times \mathbb{G} \to \mathbb{Z}_q^*$ is used to generate PPKs for vehicles, which guarantees that the SMK cannot be calculated by attackers. $H_4 : \mathbb{Z}_q^* \times \mathbb{G} \times \mathbb{G} \times \{0,1\}^* \times \{0,1\}^* \to \mathbb{Z}_q^*$ is used to generate signatures. The timestamp and public key in H_4 achieve unforgeability and resist replay attacks. $H_5 : \mathbb{Z}_q^* \times \mathbb{G} \times \mathbb{G} \times \mathbb{G} \times \mathbb{G} \to \mathbb{Z}_q^*$ is used to generate the encryption key, which supports data confidentiality in the proposed scheme.
   (4) Define a one-way index function $F_{index} : \mathbb{Z}_q^* \times \{0,1\}^* \to \mathbb{Z}_q^*$. In $F_{index}(n, PID_{R_i})$, n denotes the input number of PID_{R_i}, $PID_R = \{PID_{R_1}, PID_{R_2}, \ldots, PID_{R_n}\}$ denotes the receiver vehicle identity, and the output of $F_{index}(n, PID_{R_i})$ is

$i = 1, 2, \ldots, n$. In other words, $F_{index}(n, PID_{R_i})$ uniformly maps each user PID_{R_i} to a unique value in the set $i = 1, 2, \ldots, n$. It is used to locate the ciphertext from C_m for the receivers.

2. **KGC initialization.** In our scheme, the KGC is composed of $\{KGC_1, KGC_2, \ldots, KGC_n\}$. $KGC_i (i = 1, 2, \ldots, n)$ generate their own sub-keys and SMKs. The steps are as follows:
   (1) Randomly choose a polynomial on F_q: $g_i(x) = a_{i,0} + a_{i,1}x + \ldots + a_{i,t-1}x^{t-1}$, where $a_{i,j} \in \mathbb{Z}_q^*$.
   (2) Execute the FVSS algorithm: $FVSS(g_i(x), n, t) = (s_{i,1}, s_{i,2}, \ldots, s_{i,n})$. For $j = 1, 2, \ldots, n$, KGC_i sends $s_{i,j}$ to $KGC_j (j \neq i)$ and broadcasts commitment $\{a_{i,0}P, a_{i,1}P, \ldots, a_{i,t-1}P\}$.
   (3) Check whether $s_{j,i}P = \sum_{\ell=0}^{t-1} \left(i^{\ell}\left(a_{j,\ell}P\right)\right) + a_{j,0}P$ holds to verify the validity of $s_{j,i}$ from KGC_j. If it holds, KGC_i computes its sub-key $s_i = \sum_{j=1}^{t} s_{j,i}$ and sets sub-public key $P_i = s_iP$.
   (4) For index set $I = \{i_1, i_2, \ldots, i_t\}$, compute $\delta_{\ell} = \prod_{j \neq \ell, j \in I} \frac{j}{j-\ell}$, set SMK $s = \sum_{\ell=1}^{t} \delta_{\ell} s_{\ell}$ and system public key $P_{pub} = sP$.
   (5) Publish the system public parameters $params = \{q, P, \mathbb{G}, F_{index}(n, PID_{R_i}), F_q, T_{pub}, P_{pub}, H_0, H_1, H_2, H_3, H_4, H_5, P_i, \delta_i\} (i = 1, 2, \ldots, n)$.

4.3. Pseudonym Generation

In this stage, vehicles request registration from the TMA. In the IoV, a batch of pseudonyms is necessary to protect the privacy of vehicles' identities and historical routes. The steps are as follows:

(1) V_i randomly selects $x_i \in \mathbb{Z}_q^*$ and computes $X_i = x_iP$, $h_0 = H_0(x_iT_{pub})$, $ID_i = RID_i \oplus h_0$, and $h_{1i} = H_1(X_i||ID_i||n)$. Then, V_i sends the registration request $\{h_{1i}, X_i, ID_i, n\}$ to the TMA over public channels, where n denotes the requested number of pseudonyms.
(2) After receiving $\{h_{1i}, X_i, ID_i, n\}$, the TMA computes $h_{1i}^{'} = H_1(X_i||ID_i||n)$ and checks whether $h_{1i}^{'} = h_{1i}$. If not, the TMA rejects the registration request. Otherwise, the TMA computes $h_0 = H_0(aX_i)$ and retrieves the real identity of V_i: $RID_i = ID_i \oplus h_0$.
(3) For $j = 1, 2, \ldots, n$, the TMA computes $Q_{i,j} = RID_i \oplus h_{2i}$, where $h_{2i} = H_2(aX_i||T_{i,j})$, and sets pseudonym $PID_{i,j} = \{Q_{i,j}, T_{i,j}\}$, where $T_{i,j}$ is the valid period of $PID_{i,j}$. The anonymous identity is $PID = \{PID_{i,1}, PID_{i,2}, \ldots, PID_{i,n}\}$.
(4) To avoid heavy communication costs between vehicles and the KGC in requesting PPKs, the TMA sends PID and $\{PID, X_i\}$ to V_i and KGC, respectively.

4.4. Key Generation

In this stage, V_i obtains a PPK from the KGC and computes the complete public and private keys.

1. **PPK Generation.** After receiving $\{PID, X_i\}$, the KGC generates a PPK that corresponds to $PID_{i,j} (j = 1, 2, \ldots, n)$. To improve the robustness and reduce economic expenditure, we utilize key agreements to enable the PPK to be transmitted over public channels. The steps are as follows:
   (1) Randomly select $l_i \in \mathbb{Z}_q^*$, and compute $L_i = l_iP$, $h_{3i} = H_3(PID_{i,j}||X_i||L_i||P_{pub})$.
   (2) Set $k_i = [l_i + \delta_i s_i h_{3i} - H_0(s_iX_i)] \mod q$, and send (L_i, k_i) to V_i over public channels.
   (3) V_i computes $h_{3i} = H_3(PID_{i,j}||X_i||L_i||P_{pub})$, and checks whether $k_iP = L_i + h_{3i}\delta_iP_i - H_0(x_iP_i)P$ holds. If not, V_i applies for registration and a PPK again. Otherwise, V_i computes $y_i = [k_i + H_0(x_iP_i)] \mod q = (l_i + \delta_i s_i h_{3i}) \mod q$.

(4) For sets $y = \{y_1, y_2, \ldots, y_t\}$ and $L = \{L_1, L_2, \ldots, L_t\}$, V_i generates a complete PPK $d_i = \sum_{i=1}^{t} y_i$ and the corresponding public key $R_i = \sum_{i=1}^{t} L_i$.

2. **Private Key Generation.** Set private key $SK_i = (x_i + d_i) \bmod q$. Store SK_i and PID in a tamper-proof device (TPD).
3. **Public Key Generation.** Set public key $PK_i = (X_i, R_i)$.

4.5. Message Signcryption and Unsigncryption

In this stage, sender V_i signcrypts messages that are collected by sensors and transmits the ciphertext to other vehicles or RSUs. Receivers unsigncrypt the designated ciphertext to obtain traffic-related information. For convenience, we only describe the process of message signcryption and unsigncryption between sender V_i and receiver V_{R_i}.

1. **Message Signcryption.** This algorithm is executed by V_i. In the IoV, vehicles broadcast their public keys and communication pseudonyms on the way. Sender V_i obtains n vehicles' identities $PID_R = \{PID_{R_1}, PID_{R_2}, \ldots, PID_{R_n}\}$ and collects the corresponding messages $M = \{m_{R_1}, m_{R_2}, \ldots, m_{R_n}\}$, where $m_{R_i} \in \{0,1\}^*$. Sender V_i randomly chooses a private key SK_i and a pseudonym $PID_{i,j}$ from the TPD to signcrypt M. The steps are as follows:
   (1) Select a random integer $z \in \mathbb{Z}_q^*$ and compute $Z = zP$. For $PID_{R_i} (1 \le i \le n)$, V_i executes steps 2–4.
   (2) Compute $J_{R_i} = F_{index}(n, PID_{R_i})$, where $J_{R_i} \in [1, n]$.
   (3) Compute $h_4^{R_i} = H_4(PID_{i,j}||PK_i||Z||m_i||t_{R_i,1})$ and signature $sig_{R_i} = [z + h_4^{R_i} SK_i] \bmod q$, where $t_{R_i,1}$ is the OBU's current timestamp related to V_{R_i}. Let J_{R_i} be the index of $t_{R_i,1}$ in set T, which means that $T[J_{R_i}] = t_{R_i,1}$.
   (4) Compute $U_{R_i} = zX_{R_i}$, $K_{R_i} = H_5(PID_{i,j}||PK_i||Z||PK_{R_i}||U_{R_i})$ and ciphertext $c_{R_i} = K_{R_i} \oplus (m_i||sig_{R_i})$. Let J_{R_i} be the index of c_{R_i} in CT, which means that $CT[J_{R_i}] = c_{R_i}$.
   (5) Set $C_m = \{Z, T, CT\}$ as the signcryption ciphertext and send it to all receivers.
2. **Message Unsigncryption.** This algorithm is executed by receiver V_{R_i}. The steps are as follows:
   (1) Compute $J_{R_i} = F_{index}(n, PID_{R_i})$.
   (2) Obtain $t_{R_i,1}$ from T and check whether $|t_{R_i,2} - t_{R_i,1}| \le \Delta tt$ holds, where $t_{R_i,2}$ denotes the current timestamp of V_{R_i}, and Δtt denotes the valid time interval. If so, V_{R_i} executes step 3. Otherwise, output $\perp$.
   (3) Obtain c_{R_i} from CT and compute $U'_{R_i} = x_{R_i} Z$, $K'_{R_i} = H_5(PID_{i,j}||PK_i||Z||PK_{R_i}||U'_{R_i})$, $(m_{R_i}||sig_{R_i}) = K'_{R_i} \oplus c_{R_i}$.
   (4) Compute $h_{3i} = H_3(PID_{i,j}||X_i||R_i||P_{pub})$, $h_4^{R_i} = H_4(PID_{i,j}||PK_i||Z||m_{R_i}||\ t_{R_i,1})$, and check whether $sig_{R_i} P = Z + h_4^{R_i}(X_\mathrm{i} + R_\mathrm{i} + h_{3i} P_{pub})$ holds. If not, output $\perp$. Otherwise, output m_{R_i}.

4.6. KGC Secret Key Update

In this stage, $KGC_i (i = 1, 2, \ldots, n)$ update their sub-keys and SMKs to resist APT attacks. The steps are as follows:

(1) Randomly choose a polynomial on $F_q : g_i^x(x) = b_{i,0} + b_{i,1}x + \ldots + b_{i,t-1}x^{t-1}$, where $b_{i,j} \in \mathbb{Z}_q^*$.
(2) Execute the FVSS algorithm: $FVSS(g_i^x(x), n, t) = (w_{i,1}, w_{i,2}, \ldots, w_{i,n})$. For $j = 1, 2, \ldots, n$, KGC_i sends $w_{i,j}$ to KGC_j $(j \ne i)$ and broadcasts commitment $\{b_{i,0}P, b_{i,1}P, \ldots, b_{i,t-1}P\}$.

(3) Check whether $w_{j,i}P = \sum_{\ell=0}^{t-1}\left(i^{\ell}\left(b_{j,\ell}P\right)\right) + b_{j,0}P$ holds to verify the validity of $w_{j,i}$ from KGC_j $(j = 1, 2, \ldots, n, j \neq i)$. If it holds, KGC_i computes its new sub-key $s_i^{'} = \sum_{j=1}^{t} w_{j,i}$ and sets a new sub-public key $P_i^{'} = s_i^{'}P$.

(4) For index set $I = \{i_1, i_2, \ldots, i_t\}$, compute $\delta_\ell = \prod_{j \neq \ell, j \in I} \frac{j}{j-\ell}$ and set a new SMK $s^{'} = \sum_{\ell=1}^{t} \delta_\ell s_\ell^{'}$ and a new system public key $P_{pub}^{'} = s^{'}P$.

(5) After the KGC finishes the key update, it instructs vehicles to reapply for a PPK.

4.7. Malicious Vehicle Tracing

In this stage, the TMA traces and punishes malicious vehicles that transmit fake or faulty messages. The steps are as follows:

(1) If vehicles in the IoV find that the message is fake or faulty after decryption, they send the sender vehicle's $PID_{i,j}$ and detailed illegal behavior to the TMA with the help of the RSUs.

(2) When receiving $PID_{i,j} = \{Q_{i,j}, T_{i,j}\}$, the TMA first quickly retrieves $RID_i = Q_{i,j} \oplus H_2(aX_i||T_{i,j})$ with private key a. Then, the TMA investigates and verifies the reporting information. If it is true, the TMA reduces the reputation level of vehicle RID_i and imposes fines on its owners. Moreover, the punished vehicle may be removed from the IoV assuming that RID_i made a major mistake. Lastly, the TMA sets the vehicle's corresponding public key X_i as invalid and broadcasts this to the RSUs. The RSUs stop supporting services for RID_i for a period of time.

4.8. Correctness Analysis of the LS-MRCLSC Scheme

The correctness of the LS-MRCLSC scheme is guaranteed by the following equations.

1. From Sections 4.2 and 4.4, we know that $P_i = s_iP$, $L_i = l_iP$, $X_i = x_iP$, and $k_i = r_i + \delta_i s_i h_{3i} - H_0(s_iX_i)$. Since vehicle V_i does not know s_i, it computes $H_0(s_iX_i) = H_0(s_ix_iP) = H_0(x_i(s_iP)) = H_0(x_iP_i)$ according to the key agreement in [47]. Then, V_i verifies the correctness of the temporary PPK using the following equation:

$$\begin{aligned} k_iP &= (l_i + h_{3i}\delta_i s_i - H_0(x_iP_i))P \\ &= l_iP + h_{3i}\delta_i(s_iP) - H_0(x_iP_i)P \\ &= L_i + \delta_i h_{3i}P_i - H_0(x_iP_i)P \end{aligned} \quad (1)$$

2. From Section 4.5, we know that $U_{R_i} = zX_{R_i}$, $X_{R_i} = x_{R_i}P$, $c_{R_i} = K_{R_i} \oplus (m_{R_i}||sig_{R_i})$. After computing index J_{R_i}, receiver vehicle V_{R_i} obtains parameter Z. Since V_{R_i} does not know z, it computes $U_{R_i}^{'} = zX_{R_i} = zx_{R_i}P = x_{R_i}(zP) = x_{R_i}Z$ according to the key agreement in [47]. If V_{R_i} can decrypt c_{R_i} with $K_{R_i}^{'} = H_5(Z||PK_{R_i}||U_{R_i}^{'}) = K_{R_i}$, it means that Z has not been tampered with or replaced. The correctness of the decryption process is expressed as follows:

$$\begin{aligned} (m_{R_i}||sig_{R_i}) &= K_{R_i}^{'} \oplus c_{R_i} \\ &= H_5(PID_{i,j}||PK_i||Z||PK_{R_i}||U_{R_i}^{'}) \oplus c_{R_i} \\ &= H_5(PID_{i,j}||PK_i||Z||PK_{R_i}||x_{R_i}Z) \oplus c_{R_i} \\ &= H_5(PID_{i,j}||PK_i||Z||PK_{R_i}||zX_{R_i}) \oplus c_{R_i} \\ &= H_5(PID_{i,j}||PK_iZ||PK_{R_i}||U_{R_i}) \oplus c_{R_i} \\ &= K_{R_i} \oplus c_{R_i} \end{aligned} \quad (2)$$

3. From Section 4.4, we know that $SK_i = (x_i + d_i)$, $R_i = \sum_{i=1}^{t} L_i$, $d_i = \sum_{i=1}^{t} y_i = \sum_{i=1}^{t} (l_i + \delta_i s_i h_{3i}) = \sum_{i=1}^{t} l_i + \sum_{i=1}^{t} (\delta_i s_i h_{3i})$, $X_i = x_i P$, $P_{pub} = sP = \sum_{i=1}^{t} (\delta_i s_i P)$, $d_i P = \sum_{i=1}^{t} y_i P = \sum_{i=1}^{t} (l_i + \delta_i s_i h_{3i}) P = \sum_{i=1}^{t} l_i P + \sum_{i=1}^{t} (\delta_i s_i h_{3i}) P = \sum_{i=1}^{t} L_i + h_{3i} P_{pub} = R_i + h_{3i} P_{pub}$. Therefore, the correctness of the verification process is expressed as follows:

$$\begin{aligned} sig_{R_i} P &= (z + h_4^{R_i} SK_i) P \\ &= (z + h_4^{R_i} (x_i + d_i)) P \\ &= zP + h_4^{R_i} (x_i P + d_i P) \\ &= Z + h_4^{R_i} (X_i + R_i + h_{3i} P_{pub}) \end{aligned} \tag{3}$$

5. Security Analysis

In this section, we prove that our LS-MRCLSC scheme can withstand malicious V_i attacks and malicious KGC attacks through Theorems 1–4.

5.1. Data Confidentiality

Theorem 1. *Confidentiality against a Type I attack. If IND-CCA2 adversary $A_{I\text{-}1}$ can win Game I with a non-negligible probability $\mathcal{E}$ in polynomial time, then challenger $C_{I\text{-}1}$ has the advantage of $\left(\frac{1}{q_1}\right)^2 \frac{\mathcal{E}}{q_2 q_3}$ in solving the ECCDHP.*

Proof of Theorem 1. Assume that challenger $C_{\text{I-1}}$ receives a random example of the EC-CDHP (p, q, P, aP, bP), where $a, b \in \mathbb{Z}_q^*$ and a, b are unknown. The goal of $C_{\text{I-1}}$ is to calculate abP. $C_{\text{I-1}}$ needs the ability of $A_{\text{I-1}}$ and plays the role of a challenger in the IND-CCA2 game.

Setup: $C_{\text{I-1}}$ executes the system initialization algorithm and sends $params = \{q, P, G, F_{index}, F_q, T_{pub}, P_{pub}, H_0, H_1, H_2, H_3, H_4, H_5, P_i\} (i = 1, 2, \ldots, n)$ to $A_{\text{I-1}}$. $C_{\text{I-1}}$ maintains the list $L_0, L_1, L_2, L_3, L_4, L_5, L_P, L_{Pri}, L_{Pub}$, which is used to record the results of the H_0 query, H_1query, H_2 query, H_3 query, H_4 query, H_5 query, PPK generation query, private key query, and public key query, respectively. Initialize all lists as null. The interactive process between $A_{\text{I-1}}$ and $C_{\text{I-1}}$ is as follows.

Query Stage: $A_{\text{I-1}}$ executes the following queries adaptively.

- H_0 query: When $A_{\text{I-1}}$ queries $\{aX_i, h_0\}$, $C_{\text{I-1}}$ returns it if it exists in L_0. Otherwise, $C_{\text{I-1}}$ randomly chooses $h_0 \in \mathbb{Z}_q^*$, adds $\{aX_i, h_0\}$ to L_0, and returns h_0 to $A_{\text{I-1}}$.
- H_1 query: When $A_{\text{I-1}}$ queries $\{X_i, ID_i, n, h_1\}$, $C_{\text{I-1}}$ returns it if it exists in L_1. Otherwise, $C_{\text{I-1}}$ randomly chooses $h_1 \in \mathbb{Z}_q^*$, adds $\{X_i, ID_i, n, h_1\}$ to L_1, and returns h_1 to $A_{\text{I-1}}$.
- H_2 query: When $A_{\text{I-1}}$ queries $\{aX_i, T_{i,j}, h_{2i}\}$, $C_{\text{I-1}}$ returns it if it exists in L_2. Otherwise, $C_{\text{I-1}}$ randomly chooses $h_{2i} \in \mathbb{Z}_q^*$, adds $\{aX_i, T_{i,j}, h_{2i}\}$ to L_2, and returns h_{2i} to $A_{\text{I-1}}$.
- H_3 query: When $A_{\text{I-1}}$ queries $\{PID_{i,j}, X_i, R_i, P_{pub}, h_{3i}\}$, $C_{\text{I-1}}$ returns it if it exists in L_3. Otherwise, $C_{\text{I-1}}$ selects $c \in \{0, 1\}$, where $Pr[c = 1] = \delta$. When $c = 0$, $C_{\text{I-1}}$ randomly chooses $h_{3i} \in \mathbb{Z}_q^*$, adds $\{PID_{i,j}, X_i, R_i, P_{pub}, h_{3i}\}$ to L_3, and returns h_{3i} to $A_{\text{I-1}}$.
- H_4 query: When $A_{\text{I-1}}$ queries $\{PID_{i,j}, PK_i, Z, m_{R_i}, t_{R_i}, h_4^{R_i}\}$, $C_{\text{I-1}}$ returns it if it exists in L_4. Otherwise, $C_{\text{I-1}}$ randomly chooses $h_4^{R_i} \in \mathbb{Z}_q^*$, adds $\{PID_{i,j}, PK_i, Z, m_{R_i}, t_{R_i}, h_4^{R_i}\}$ to L_4, and returns $h_4^{R_i}$ to $A_{\text{I-1}}$.
- H_5 query: When $A_{\text{I-1}}$ queries $\{PID_{i,j}, PK_i, Z, PK_{R_i}, U_{R_i}, K_{R_i}\}$, $C_{\text{I-1}}$ returns it if it exists in L_5. Otherwise, $C_{\text{I-1}}$ randomly chooses $K_{R_i} \in \mathbb{Z}_q^*$, adds $\{PID_{i,j}, PK_i, Z, PK_{R_i}, U_{R_i}, K_{R_i}\}$ to L_5, and returns K_{R_i} to $A_{\text{I-1}}$.

PPK generation query: When $A_{\text{I-1}}$ queries $\{PID_{i,j}, R_i, d_i\}$, $C_{\text{I-1}}$ returns it if it exists in L_P. Otherwise, $C_{\text{I-1}}$ looks up $\left\{PID_{i,j}, X_i, R_i, P_{pub}\right\} \in L_3$, selects a random integer $d_i \in \mathbb{Z}_q^*$, calculates $R_i = d_iP - h_{3i}P_{pub}$, returns (d_i, R_i) to $A_{\text{I-1}}$, and adds $\{PID_{i,j}, R_i, d_i\}$ to L_P.

Private key generation query: When $A_{\text{I-1}}$ queries $\{PID_{i,j}, SK_i\}$, $C_{\text{I-1}}$ returns it if it exists in L_{Pri}. Otherwise, $C_{\text{I-1}}$ looks for L_P to obtain d_i and randomly selects $x_i \in \mathbb{Z}_q^*$, returns $\{PID_{i,j}, SK_i\}$ to $A_{\text{I-1}}$, and adds it to L_{Pri}.

Public key generation query: When $A_{\text{I-1}}$ queries $\{PID_{i,j}, PK_i = (X_i, R_i)\}$, $C_{\text{I-1}}$ returns it if it exists in L_{Pub}. Otherwise, $C_{\text{I-1}}$ looks for L_P, calculates $R_i = d_iP - h_{3i}P_{pub}$, selects $x_i \in \mathbb{Z}_q^*$, computes $X_i = x_iP$, $PK_i = (X_i, R_i)$, adds $\{PID_{i,j}, PK_i = (X_i, R_i)\}$ to L_{Pub}, and returns it to $A_{\text{I-1}}$. If no corresponding records exist in L_P, then it looks for L_3. If $c = 1$, $C_{\text{I-1}}$ selects two random integers $x_i, d_i \in \mathbb{Z}_q^*$, calculates $X_i = x_iP$, and $R_i = d_iP - h_{3i}P_{pub}$; sets $PK_i = (X_i, R_i)$; adds $\{PID_{i,j}, PK_i = (X_i, R_i)\}$ to L_{Pub}; and returns it to $A_{\text{I-1}}$. If $c = 0$, $C_{\text{I-1}}$ performs a PPK generation query to obtain (d_i, R_i), elects $x_i \in \mathbb{Z}_q^*$, computes $X_i = x_iP$, sets $PK_i = (X_i, R_i)$, adds $\{PID_{i,j}, PK_i = (X_i, R_i)\}$ to L_{Pub}, and returns it to $A_{\text{I-1}}$.

Public key replacement: $A_{\text{I-1}}$ chooses a new $sig_{R_i}^{'}$ to replace sig_{R_i} and replaces PK_i with a new $PK_i^{'}$. $C_{\text{I-1}}$ updates L_{Pub} with $\left\{PID_{i,j}, PK_i^{'}\right\}$.

For convenience, we assume that the sender is PID_a and the receiver is PID_b.

Signcryption query: When $A_{\text{I-1}}$ queries $\{PID_a, PID_b, sig_{R_b}, c, m_{R_b}\}$, $C_{\text{I-1}}$ first looks for $\{PID_b, PK_{R_b} = (X_{R_b}, R_{R_b}), c\}$ in L_{Pub}. If $c = 1$, $C_{\text{I-1}}$ aborts the game. Otherwise, $C_{\text{I-1}}$ looks for $\{PID_a, SK_a\}$ in L_{Pri} and $\{PID_a, PK_a = (X_a, R_a)\}$. $C_{\text{I-1}}$ randomly selects $z \in \mathbb{Z}_q^*$; computes $Z = zP$, $h_4^{R_b} = H_4(PID_a||PK_a||Z||m_{R_b}||t_{R_b,1})$, $sig_{R_b} = \left[z + h_4^{R_b}SK_a\right] \bmod q$, $U_{R_b} = zX_{R_b}$, $K_{R_b} = H_5\left(PID_a||PK_a||Z||PK_{R_b}||U_{R_b}\right)$, and $c_{R_b} = K_{R_b} \oplus (m_{R_b}||sig_{R_b})$; and returns $C_m = \{c_{R_b}, t_{R_b}, sig_{R_b}\}$ to $A_{\text{I-1}}$.

Unsigncryption query: When $A_{\text{I-1}}$ queries $\{PID_a, PID_b, C_m, m_{R_b}\}$, $C_{\text{I-1}}$ first looks for PID_a in L_{Pub}.

(1) If PID_a exists and $c = 0$, then $C_{\text{I-1}}$ looks for $\{PID_b, x_b\}$, $\{PID_a, PK_a = (X_a, R_a)\}$, and $\left\{PID_b, PK_{R_b} = (X_{R_b}, R_{R_b})\right\}$ in L_{Pri} and L_{Pub}; computes $U_{R_b}^{'} = x_bZ$, $K_{R_b}^{'} = H_5(PID_a||PK_a||Z||PK_{R_b}||U_{R_b}^{'})$, and $(m_{R_b}||sig_{R_b}) = K_{R_b}^{'} \oplus c_{R_b}$; and returns m_{R_b}. Otherwise, $C_{\text{I-1}}$ aborts.

(2) If PID_a exists and $c = 1$, then $C_{\text{I-1}}$ looks for $\{PID_a, PK_a = (X_a, R_a)\}$ in L_{Pub}, $\{PID_a, PK_a, Z, m_{R_n}, t_{R_b}, h_4^{R_b}\}$ in L_4, and $\left\{PID_b, Z, PK_b, U_{R_b}, K_{R_b}\right\}$ in L_5 and returns m_{R_b}. Otherwise, $C_{\text{I-1}}$ aborts.

(3) If PID_a does not exist in L_{Pub} (the public key has been replaced), $C_{\text{I-1}}$ looks for $\{PID_a, PK_a = (X_a, R_a)\}$ in L_{Pub}, $\left\{PID_a, PK_a, Z, m_{R_n}, t_{R_b}, h_4^{R_b}\right\}$ in L_4, and $\left\{PID_b, Z, PK_b, U_{R_b}, K_{R_b}\right\}$ in L_5 and returns m_{R_b}. Otherwise, $C_{\text{I-1}}$ aborts.

Challenge Stage: After polynomial-bounded degree queries, $A_{\text{I-1}}$ outputs two identities $\{PID_a, PID_b\}$ and two messages $\{m_0, m_1\}$ as a challenge. If $c = 0$, $C_{\text{I-1}}$ aborts. Otherwise, it randomly selects $z^*, h_4^* \in \mathbb{Z}_q^*$ and computes $j \in \{0, 1\}$, $sig_{R_b}^* = z^* + h_4^*SK_a$, $Z^* = z^*P$, $K_{R_i} = H_5\left(PID_a||PK_a||Z^*||PK_b||U_{R_b}\right)$, and $c_{R_b}^* = K_{R_b} \oplus (m_j||sig_{R_b}^*)$. Then, $C_{\text{I-1}}$ submits the challenge ciphertext $C_m^* = \{c_{R_b}^*, t_{R_b}^*, sig_{R_b}^*\}$ to $A_{\text{I-1}}$, where $C_{\text{I-1}}$ knows the information of the public key replacement.

Guess Stage: $A_{\text{I-1}}$ continues polynomial-bounded degree queries, and outputs $j^{'}$ as the guess of j when the simulation stops. If $j^{'} = j$, $C_{\text{I-1}}$ outputs $(\frac{SK_b - x_b}{h_{3i}})Z^* - \frac{z^*}{h_{3i}}R_b = sZ^*$ as the solution of the ECCDHP. Otherwise, $C_{\text{I-1}}$ fails.

Now, we analyze the probability that $C_{\text{I-1}}$ outputs the correct solution of the ECCDHP. If the following two conditions are satisfied, $A_{\text{I-1}}$ wins Game I.

(1) $A_{\text{I-1}}$ did not submit a PPK generation query or private key query, whose probability is $\left(\frac{1}{q_1}\right)^2$.

(2) $A_{\text{I-1}}$ did not execute an H_4 query or H_5 query with the probability $\frac{1}{q_2 q_3}$.

In summary, if $A_{\text{I-1}}$ wins Game I with a non-negligible probability $\mathcal{E}$ in polynomial time, $C_{\text{I-1}}$ can solve the ECCDHP with the probability $\left(\frac{1}{q_1}\right)^2 \frac{\mathcal{E}}{q_2 q_3}$. □

Theorem 2. *Confidentiality against a Type II attack. If IND-CCA2 adversary $A_{II\text{-}1}$ can win Game II with a non-negligible probability $\mathcal{E}$ in polynomial time, then challenger $C_{II\text{-}1}$ has the advantage of $\frac{\mathcal{E}}{q_1 q_2 q_3}$ in solving the ECCDHP.*

Proof of Theorem 2. Assume that challenger $C_{\text{II-1}}$ receives a random example of the ECCDHP (p, q, P, aP, bP), where $a, b \in \mathbb{Z}_q^*$ and a, b are unknown. The goal of $C_{\text{II-1}}$ is to calculate abP. $C_{\text{II-1}}$ needs the ability of $A_{\text{II-1}}$ and plays the role of a challenger in the IND-CCA2 game.

Setup: $C_{\text{II-1}}$ executes the system initialization algorithm and sends $params = \{q, P, G, F_{index}, F_q, T_{pub}, P_{pub}, H_0, H_1, H_2, H_3, H_4, H_5, P_i\}(i = 1, 2, \ldots, n)$ to $A_{\text{II-1}}$. $A_{\text{II-1}}$ knows the SMK s but cannot execute public key replacement attacks and PPK generation queries. The other assumptions are the same as Theorem 1. The interactive process between $A_{\text{II-1}}$ and $C_{\text{II-1}}$ is as follows.

Query Stage: $A_{\text{II-1}}$ executes an H_0 query, H_1 query, H_2 query, H_3 query, H_4 query, H_5 query, private key generation query, public key generation query, and signcryption query adaptively.

For convenience, we assume that the sender is PID_a and the receiver is PID_b.

Unsigncryption query: When $A_{\text{II-1}}$ queries $\{PID_a, PID_b, C_m, m_{R_b}\}$, $C_{\text{II-1}}$ first looks for PID_a in L_{Pub}.

(1) If PID_a exists and $c = 0$, then $C_{\text{II-1}}$ looks for $\{PID_b, x_b\}$, $\{PID_a, PK_a = (X_a, R_a)\}$, and $\left\{PID_b, PK_{X_{R_b}} = (X_{R_b}, R_{R_b})\right\}$ in L_{Pri} and L_{Pub}; computes $U_{R_b}^{'} = x_b Z$, $K_{R_b}^{'} = H_5(PID_a||PK_a||Z||PK_{R_b}||U_{R_b}^{'})$, an $(m_{R_b}||sig_{R_b}) = K_{R_b}^{'} \oplus c_{R_b}$; and returns m_{R_b}. Otherwise, $C_{\text{II-1}}$ aborts.

(2) If PID_a exists and $c = 1$, then $C_{\text{II-1}}$ looks for $\{PID_a, PK_a = (X_a, R_a)\}$ in L_{Pub}, $\{PID_{i,j}, PK_i, Z, m_{R_i}, t_{R_i}, h_4^{R_i}\}$ in L_4, and $\{Z, PK_i, U_{R_i}, K_{R_i}\}$ in L_5 and returns m_{R_b}. Otherwise, $C_{\text{II-1}}$ aborts.

(3) If PID_a does not exist in L_{Pub} (the public key has been replaced), $C_{\text{II-1}}$ looks for $\{PID_a, PK_a = (X_a, R_a)\}$ in L_{Pub}, $\left\{PID_{i,j}, PK_i, Z, m_{R_i}, t_{R_i}, h_4^{R_i}\right\}$ in L_4, and $\{Z, PK_i, U_{R_i}, K_{R_i}\}$ in L_5 and returns m_{R_b}. Otherwise, $C_{\text{II-1}}$ aborts.

Challenge Stage: After polynomial-bounded degree queries, $A_{\text{II-1}}$ outputs two identities $\{PID_a, PID_b\}$ and two messages $\{m_0, m_1\}$ as a challenge. If $c = 0$, $C_{\text{II-1}}$ aborts. Otherwise, it randomly selects $z^*, h_4^* \in \mathbb{Z}_q^*$ and computes $j \in \{0, 1\}$, $Z^* = z^* P$, $sig_{R_b}^* = z^* + h_4^* SK_a$, $K_{R_i} = H_5(PID_a||PK_a||Z^*||PK_b||U_{R_b})$, and $c_{R_b}^* = K_{R_b} \oplus (m_j||sig_{R_b}^*)$. Then, $C_{\text{II-1}}$ submits the challenge ciphertext $C_m^* = \{c_{R_b}^*, t_{R_b}^*, sig_{R_b}^*\}$ to $A_{\text{II-1}}$, where $C_{\text{II-1}}$ knows the SMK s.

Guess Stage: $A_{\text{II-1}}$ continues polynomial-bounded degree queries and outputs $j^{'}$ as the guess of j when the simulation stops. If $j^{'} = j$, $C_{\text{II-1}}$ outputs $(SK_b - d_b)Z^* = x_b Z^*$ as the solution of the ECCDHP. Otherwise, $C_{\text{II-1}}$ fails.

Now, we analyze the probability that $C_{\text{II-1}}$ outputs the correct solution of the ECCDHP. If the following two conditions are satisfied, $A_{\text{II-1}}$ wins Game II.

(1) $A_{\text{II-1}}$ did not submit a private key query, whose probability is $\frac{1}{q_1}$.

(2) $A_{\text{II-1}}$ did not execute an H_4 query or H_5 query with the probability $\frac{1}{q_2 q_3}$.

In summary, if $A_{\text{II-1}}$ wins Game I with a non-negligible probability $\mathcal{E}$ in polynomial time, $C_{\text{II-1}}$ can solve the ECCDHP with the probability $\frac{\mathcal{E}}{q_1 q_2 q_3}$. □

5.2. Message Unforgeability

Theorem 3. *Unforgeability against a Type I attack. If EUF-CMA adversary $A_{I\text{-}2}$ can win Game III with a non-negligible probability $\mathcal{E}$ in polynomial time, then challenger $C_{I\text{-}2}$ has the advantage of $\left(\frac{1}{q_1}\right)^2 \frac{\mathcal{E}}{q_2}$ in solving the ECDLP.*

Proof of Theorem 3. Assume that challenger $C_{\text{I-2}}$ receives a random example of the ECDLP (p, q, P, aP), where $a \in \mathbb{Z}_q^*$ and a is unknown. The goal of $C_{\text{I-2}}$ is to calculate a. $C_{\text{I-2}}$ needs the ability of $A_{\text{I-2}}$ and plays the role of a challenger in the EUF-CMA game.

Setup: $C_{\text{I-2}}$ executes the system initialization algorithm and sends $params = \{q, P, G, F_{index}, F_q, T_{pub}, P_{pub}, H_0, H_1, H_2, H_3, H_4, H_5, P_i\}(i = 1, 2, \ldots, n)$ to $A_{\text{I-2}}$. $C_{\text{I-2}}$ maintains the list $L_0, L_1, L_2, L_3, L_4, L_5, L_P, L_{Pri}, L_{Pub}$, which is used to record the results of the H_0 query, H_1query, H_2 query, H_3 query, H_4 query, H_5 query, PPK generation query, private key query, and public key query, respectively. Initialize all lists as null. The interactive process between $A_{\text{I-2}}$ and $C_{\text{I-2}}$ is as follows.

Query Stage: $A_{\text{I-2}}$ executes the following queries adaptively.

- H_0 query: When $A_{\text{I-2}}$ queries $\{aX_i, h_0\}$, $C_{\text{I-2}}$ returns it if it exists in L_0. Otherwise, $C_{\text{I-2}}$ randomly chooses $h_0 \in \mathbb{Z}_q^*$, adds $\{aX_i, h_0\}$ to L_0, and returns h_0 to $A_{\text{I-2}}$.
- H_1 query: When $A_{\text{I-2}}$ queries $\{X_i, ID_i, n, h_1\}$, $C_{\text{I-2}}$ returns it if it exists in L_1. Otherwise, $C_{\text{I-2}}$ randomly chooses $h_1 \in \mathbb{Z}_q^*$, adds $\{X_i, ID_i, n, h_1\}$ to L_1, and returns h_1 to $A_{\text{I-2}}$.
- H_2 query: When $A_{\text{I-2}}$ queries $\{aX_i, T_{i,j}, h_{2i}\}$, $C_{\text{I-2}}$ returns it if it exists in L_2. Otherwise, $C_{\text{I-2}}$ randomly chooses $h_{2i} \in \mathbb{Z}_q^*$, adds $\{aX_i, T_{i,j}, h_{2i}\}$ to L_2, and returns h_{2i} to $A_{\text{I-2}}$.
- H_3 query: When $A_{\text{I-2}}$ queries $\left\{PID_{i,j}, X_i, R_i, P_{pub}, h_{3i}\right\}$, $C_{\text{I-2}}$ returns it if it exists in L_3. Otherwise, $C_{\text{I-2}}$ selects $c \in \{0, 1\}$, where $Pr[c = 1] = \delta$. When $c = 0$, $C_{\text{I-2}}$ randomly chooses $h_{3i} \in \mathbb{Z}_q^*$, adds $\left\{PID_{i,j}, X_i, R_i, P_{pub}, h_{3i}\right\}$ to L_3, and returns h_{3i} to $A_{\text{I-2}}$.
- H_4 query: When $A_{\text{I-2}}$ queries $\left\{PID_{i,j}, PK_i, Z, m_{R_i}, t_{R_i}, h_4^{R_i}\right\}$, $C_{\text{I-2}}$ returns it if it exists in L_4. Otherwise, $C_{\text{I-2}}$ randomly chooses $h_4^{R_i} \in \mathbb{Z}_q^*$, adds $\left\{PID_{i,j}, PK_i, Z, m_{R_i}, t_{R_i}, h_4^{R_i}\right\}$ to L_4, and returns $h_4^{R_i}$ to $A_{\text{I-2}}$.

PPK generation query: When $A_{\text{I-2}}$ queries $\left\{PID_{i,j}, R_i, d_i\right\}$, $C_{\text{I-2}}$ returns it if it exists in L_P. Otherwise, $C_{\text{I-2}}$ looks up $\left\{PID_{i,j}, X_i, R_i, P_{pub}\right\} \in L_3$, selects a random integer $d_i \in \mathbb{Z}_q^*$, calculates $R_i = d_i P - h_{3i} P_{pub}$, returns (d_i, R_i) to $A_{\text{I-2}}$, and adds $\left\{PID_{i,j}, R_i, d_i\right\}$ to L_P.

Private key generation query: When $A_{\text{I-2}}$ queries $\left\{PID_{i,j}, SK_i\right\}$, $C_{\text{I-2}}$ returns it if it exists in L_{Pri}. Otherwise, $C_{\text{I-2}}$ looks for L_P to obtain d_i, randomly selects $x_i \in \mathbb{Z}_q^*$, returns $\left\{PID_{i,j}, SK_i\right\}$ to $A_{\text{I-2}}$, and adds it to L_{Pri}.

Public key generation query: When $A_{\text{I-2}}$ queries $\left\{PID_{i,j}, PK_i = (X_i, R_i)\right\}$, $C_{\text{I-2}}$ returns it if it exists in L_{Pub}. Otherwise, $C_{\text{I-2}}$ looks for L_P, calculates $R_i = d_i P - h_{3i} P_{pub}$, selects $x_i \in \mathbb{Z}_q^*$, computes $X_i = x_i P$, $PK_i = (X_i, R_i)$, adds $\left\{PID_{i,j}, PK_i = (X_i, R_i)\right\}$ to L_{Pub}, and returns it to $A_{\text{I-2}}$. If no corresponding records exist in L_P, then it looks for L_3. If $c = 1$, $C_{\text{I-2}}$ selects two random integers $x_i, d_i \in \mathbb{Z}_q^*$; calculates $X_i = x_i P$ and $R_i = d_i P - h_{3i} P_{pub}$; sets $PK_i = (X_i, R_i)$; adds $\left\{PID_{i,j}, PK_i = (X_i, R_i)\right\}$ to L_{Pub}; and returns it to $A_{\text{I-2}}$. If $c = 0$, $C_{\text{I-2}}$ executes a PPK generation query to obtain (d_i, R_i), selects $x_i \in \mathbb{Z}_q^*$, computes $X_i = x_i P$, sets $PK_i = (X_i, R_i)$, adds $\left\{PID_{i,j}, PK_i = (X_i, R_i)\right\}$ to L_{Pub}, and returns it to $A_{\text{I-2}}$.

Public key replacement: $A_{\text{I-2}}$ chooses a new $sig_{R_i}^{'}$ to replace sig_{R_i} and replaces PK_i with a new $PK_i^{'}$. $C_{\text{I-2}}$ updates L_{Pub} with $\left\{PID_{i,j}, PK_i^{'}\right\}$.

For convenience, we assume that the sender is PID_a and the receiver is PID_b.

Sign query: When $A_{\text{I-2}}$ queries $\{PID_a, PID_b, sig_{R_b}, m_{R_b}\}$, $C_{\text{I-2}}$ first looks for $\{PID_b, PK_{R_b} = (X_{R_b}, R_{R_b}), c\}$ in L_{Pub}. If $c = 1$, $C_{\text{I-2}}$ aborts the game. Otherwise, $C_{\text{I-2}}$ looks for $\{PID_a, SK_a\}$ in L_{Pri} and $\{PID_a, PK_a = (X_a, R_a)\}$. $C_{\text{I-2}}$ randomly selects $z \in \mathbb{Z}_q^*$; com-

putes $Z = zP$, $h_4^{R_b} = H_4(PID_a||PK_a||Z||m_{R_b}||t_{R_b,1})$, and $sig_{R_b} = \left[z + h_4^{R_b} SK_a\right] \bmod q$; and returns $C_m = \{t_{R_b}, sig_{R_b}\}$ to $A_{\text{I-2}}$.

Verify query: When $A_{\text{I-2}}$ queries $\{PID_a, PID_b, C_m, m_{R_b}\}$, $C_{\text{I-2}}$ first looks for PID_a in L_{Pub}.

(1) If PID_a exists and $c = 0$, then $C_{\text{I-2}}$ looks for $\{PID_a, PK_a = (X_a, R_a)\}$ in L_{Pub} and $\{PID_a, PK_a, Z, m_{R_n}, t_{R_b}, h_4^{R_b}\}$ in L_4 and verifies that $sig_{R_b}P = Z + h_4^{R_b}(X_b + Y_b + h_{3i}P_{pub})$. If so, the output is "1". Otherwise, $C_{\text{I-2}}$ aborts.

(2) If PID_a exists and $c = 1$, then $C_{\text{I-2}}$ looks for $\{PID_a, PK_a = (X_a, R_a)\}$ in L_{Pub} and $\{PID_a, PK_a, Z, m_{R_n}, t_{R_b}, h_4^{R_b}\}$ in L_4 and verifies that $sig_{R_b}P = Z + h_4^{R_b}(X_b + Y_b + h_{3i}P_{pub})$. If so, the output is "1". Otherwise, $C_{\text{I-2}}$ aborts.

(3) If PID_a does not exist in L_{Pub} (the public key has been replaced), $C_{\text{I-2}}$ looks for $\{PID_a, PK_a = (X_a, R_a)\}$ in L_{Pub} and $\left\{PID_a, PK_a, Z, m_{R_n}, t_{R_b}, h_4^{R_b}\right\}$ in L_4 and verifies that $sig_{R_b}P = Z + h_4^{R_b}(X_b + Y_b + h_{3i}P_{pub})$. If so, the output is "1". Otherwise, $C_{\text{I-2}}$ aborts.

Forge: After polynomial-bounded degree queries, $A_{\text{I-2}}$ randomly selects $z^*, r^*, sig_{R_b}^* \in \mathbb{Z}_q^*$; obtains the current timestamp $t_{R_b}^*$; and computes $Z^* = z^*P, h_4^{R_b^*} = H_4(PID_a||PK_a||Z^*||m_i||t_{R_b}^*)$. Then, $C_{\text{I-2}}$ submits the challenge ciphertext $C_m^* = \{t_{R_b}^*, sig_{R_b}^*\}$ to $A_{\text{I-2}}$, where $C_{\text{I-2}}$ knows the information of the public key replacement. If $A_{\text{I-2}}$ successfully forges a signature, $C_{\text{I-2}}$ outputs $s = \frac{sig_{R_b}^* - z^* - h_4^{R_b^*}(x_a + r^*)}{h_{3i}h_4^{R_b^*}}$ as the solution of the ECDLP. Otherwise, $C_{\text{I-2}}$ fails.

Now, we analyze the probability that $C_{\text{I-2}}$ outputs the correct solution of the ECDLP. If the following two conditions are satisfied, $A_{\text{I-2}}$ wins Game III.

(1) $A_{\text{I-2}}$ did not submit a PPK generation query or private key query, whose probability is $\left(\frac{1}{q_1}\right)^2$.

(2) $A_{\text{I-2}}$ did not execute an H_4 query with the probability $\frac{1}{q_2}$.

In summary, if $A_{\text{I-2}}$ wins Game III with a non-negligible probability $\mathcal{E}$ in polynomial time, $C_{\text{I-2}}$ can solve the ECDLP with the probability $\left(\frac{1}{q_1}\right)^2 \frac{\mathcal{E}}{q_2}$. □

Theorem 4. *Unforgeability against a Type II attack. If EUF-CMA adversary $A_{II\text{-}2}$ can win Game IV with a non-negligible probability $\mathcal{E}$ in polynomial time, then challenger $C_{II\text{-}2}$ has the advantage of $\frac{\mathcal{E}}{q_1 q_2}$ in solving the ECDLP.*

Proof of Theorem 4. Assume challenger $C_{\text{II-2}}$ receives a random example of the ECDLP (p, q, P, aP), where $a \in \mathbb{Z}_q^*$ and a is unknown. The goal of $C_{\text{II-2}}$ is to calculate a. $C_{\text{II-2}}$ needs the ability of $A_{\text{II-2}}$ and plays the role of a challenger in the EUF-CMA game.

Setup: $C_{\text{II-2}}$ executes the system initialization algorithm and sends $params = \{q, P, G, F_{index}, F_q, T_{pub}, P_{pub}, H_0, H_1, H_2, H_3, H_4, H_5, P_i\}(i = 1, 2, \ldots, n)$ to $A_{\text{II-2}}$. $A_{\text{II-2}}$ knows the SMK s but cannot execute a public key replacement attack and PPK generation query. The other assumptions are the same as Theorem 3.

Query Stage: $A_{\text{II-2}}$ executes an H_0 query, H_1 query, H_2 query, H_3 query, H_4 query, private key generation query, public key generation query, and sign query adaptively, as in Theorem 3.

For convenience, we assume that the sender is PID_a and the receiver is PID_b.

Verify query: When $A_{\text{II-2}}$ queries $\{PID_a, PID_b, C_m, m_{R_b}\}$, $C_{\text{II-2}}$ first looks for PID_a in L_{Pub}.

(1) If PID_a exists and $c = 0$, then $C_{\text{II-2}}$ looks for $\{PID_a, PK_a = (X_a, R_a)\}$ in L_{Pub} and $\{PID_a, PK_a, Z, m_{R_n}, t_{R_b}, h_4^{R_b}\}$ in L_4 and verifies that $sig_{R_b}P = Z + h_4^{R_b}(X_b + Y_b + h_{3i}P_{pub})$. If so, the output is "1". Otherwise, $C_{\text{II-2}}$ aborts.

(2) If PID_a exists and $c = 1$, then $C_{\text{II-2}}$ looks for $\{PID_a, PK_a = (X_a, R_a)\}$ in L_{Pub} and $\{PID_a, PK_a, Z, m_{R_n}, t_{R_b}, h_4^{R_b}\}$ in L_4 and verifies that $sig_{R_b}P = Z + h_4^{R_b}(X_b + Y_b + h_{3i}P_{pub})$. If so, the output is "1". Otherwise, $C_{\text{II-2}}$ aborts.

Forge: After polynomial-bounded degree queries, $A_{\text{II-2}}$ randomly selects $z^*, r^*, sig^*_{R_b} \in \mathbb{Z}^*_q$; obtains the current timestamp $t^*_{R_b}$; and computes $Z^* = z^*P, h_4^{R^*_b} = H_4(PID_a||PK_a||Z^*||m_i||t^*_{R_b})$. Then, $C_{\text{II-2}}$ submits the challenge ciphertext $C^*_m = \{t^*_{R_b}, sig^*_{R_b}\}$ to $A_{\text{II-2}}$, where $C_{\text{II-2}}$ knows the SMK s. If $A_{\text{II-2}}$ successfully forges a signature, $C_{\text{II-2}}$ outputs $x_a = \frac{sig^*_{R_b} - z^* - h_4^{R^*_b}(r^* + sh_{3i})}{h_4^{R^*_b}}$ as the solution of the ECDLP. Otherwise, $C_{\text{II-2}}$ fails.

Now we analyze the probability that $C_{\text{II-2}}$ outputs the correct solution of the ECDLP. If the following two conditions are satisfied, $A_{\text{II-2}}$ wins Game IV.

(1) $A_{\text{II-2}}$ did not submit a private key query, whose probability is $\frac{1}{q_1}$.

(2) $A_{\text{II-2}}$ did not execute an H_4 query with the probability $\frac{1}{q_2}$.

In summary, if $A_{\text{II-2}}$ wins Game IV with a non-negligible probability $\mathcal{E}$ in polynomial time, $C_{\text{II-2}}$ can solve the ECDLP with the probability $\frac{\mathcal{E}}{q_1 q_2}$. □

5.3. Anonymity

Throughout the process of the proposed LS-MRCLSC scheme, vehicles use pseudo-identity $PID_{i,j}$ to communicate with other entities. Real identity RID_i is encrypted as $Q_{i,j} = RID_i \oplus h_{2i}$, where $PID_{i,j} = \{Q_{i,j}, T_{i,j}\}$, $h_{2i} = H_2(aX_i||T_{i,j})$, $T_{i,j}$ is the valid period of $PID_{i,j}$. To achieve the real identity of the vehicle, the adversary needs to solve the ECDLP since the adversary has to compute a satisfying $T_{pub} = aP$. Due to the intractability of the ECDLP, the proposed LS-MRCLSC scheme provides sender anonymity. Moreover, all pre-determined receiver identities $PID_R = \{PID_{R_1}, PID_{R_2}, \ldots, PID_{R_n}\}$ are not included in the ciphertext. Hence, the proposed LS-MRCLSC scheme also achieves receiver anonymity.

5.4. Unlinkability

The adversary may reveal sensitive information about the vehicle from the fixed pseudo-identity. Therefore, the proposed LS-MRCLSC scheme also provides unlinkability. Foremost, the fixed pseudo-identity $PID_{i,j}$ is replaced with a pool of pseudonyms $PID = \{PID_{i,1}, PID_{i,2}, \ldots, PID_{i,n}\}$. After signcrypting messages with different private keys, vehicle V_i chooses an unused pseudonym $PID_{i,j}$ from PID and transmits the ciphertext to the receivers. Upon finishing the last round of communication, V_i discards the used $PID_{i,j}$. Moreover, the new $PID_{i,j}$ has no relationship with the old one since $T_{i,j}$ is different in each session. Hence, the adversary cannot determine whether the senders in two or more transmission sessions are identical.

5.5. Forward and Backward Secrecy

On the one hand, in our LS-MRCLSC scheme, the private key of the vehicle consists of two parts: $SK_i = (x_i + d_i)$, where x_i is a random secret value selected by vehicles and d_i is the PPK generated by the KGC. Concretely, the PPK $d_i == \sum_{i=1}^{t} y_i = \sum_{i=1}^{t}(l_i + \delta_i s_i h_{3i})$, where l_i is a random secret value, s_i is the sub-key of each KGC, $h_{3i} = H_3(PID_{i,j}||X_i||L_i||P_{pub})$, $X_i = x_iP$, $R_i = \sum_{i=1}^{t} L_i$, and $P_i = s_iP$. Because l_i and $PID_{i,j}$ are different in each epoch, the PPK d_i and private key SK_i are different in each session. Hence, the adversary cannot obtain the previous or subsequent session keys even if the current session key SK_i has been disclosed. On the other hand, each KGC_i periodically updates its own sub-key, leading to updates in the PPK. Therefore, even if the adversary compromised t KGC_i in one epoch during a period of non-update, it would not last too long. Upon timer triggers, each KGC_i

updates its own sub-key. Thus, our proposed LS-MRCLSC scheme provides both forward and backward secrecy.

5.6. Resist KGC Damage Attacks

A compromised key server is an adversary and can extract the SMK. Several compromised KGCs can even launch more severe attacks in collusion. For the security of KGCs in our proposed scheme, each KGC_i $(1 \leq i \leq n)$ should update its secret key during each epoch. When a participant tries to obtain the SMK, it has to collect at least t secret shares from those KGCs. Based on the assumption in Section 3, we suggest that an adversary who collects t secret shares at different epochs cannot reconstruct the SMK s. Briefly, we assume that t key generation centers are broken through in two successive epochs. The adversary can obtain t shares of s_i, which are $\{s_1^{\psi}, s_2^{\psi}, \ldots, s_k^{\psi}\}$ at the ψ-th epoch and $\{s_{k+1}^{\psi+1}, s_{k+2}^{\psi+1}, \ldots, s_t^{\psi+1}\}$ at the $(\psi+1)$-th epoch.

5.7. Resist Replay Attacks

In our LS-MRCLSC scheme, timestamp $t_{R_i,1}$ is used to guarantee the freshness of ciphertext C_m, which can effectively resist replay attacks. If the adversary replays ciphertext C_m, it cannot pass authentication because of an invalid timestamp $t_{R_i,1}$. Specifically, during the message signcryption and unsigncryption stage, we assume that the predefined threshold of the period is Δtt, and the time each receiver vehicle receives $C_m = \{Z, T, CT\}$ is $t_{R_i,2}$. If $|t_{R_i,1} - t_{R_i,2}| \leq \Delta tt$, then C_m reaches receiver vehicle V_{R_i} within a valid time interval. Otherwise, receiver V_{R_i} regards C_m as a revised message and discards it. Thus, our LS-MRCLSC scheme can resist replay attacks.

Next, a security comparison between existing schemes [7–11] and our LS-MRCLSC scheme is presented in Table 2, where "✓" represents satisfying the property and "×" represents not satisfying the property.

Table 2. Security comparison.

Scheme	Scheme [7]	Scheme [8]	Scheme [9]	Scheme [10]	Scheme [11]	Ours
Data confidentiality	✓	✓	✓	✓	✓	✓
Message unforgeability	✓	✓	✓	✓	✓	✓
Anonymity	✓	✓	×	✓	✓	✓
Unlinkability	×	×	×	×	×	✓
Resist KGC damage attacks	×	×	×	×	×	✓
Forward and backward secrecy	✓	×	×	×	×	✓
Resist replay attacks	✓	×	×	×	×	✓
Without secure channels	×	×	×	✓	×	✓

In Peng's scheme [9], the sender's anonymity cannot be achieved. Apart from Ming's scheme [10], the schemes in [7–9] require secure channels during the PPK generation, but their robustness is weak. In addition, in the schemes [8–10], users utilize a fixed private key for a long time, which makes the system vulnerable to attacks. Moreover, the schemes in [7–11] utilize only one KGC, so they cannot resist KGC damage attacks and avoid SPoFs. Meanwhile, users utilize one identity to communicate with others, so unlinkability is not satisfied. Our scheme meets all security requirements, which is more practical.

6. Performance Evaluation

In Sections 4.8 and 5, the correctness and security of the proposed LS-MRCLSC scheme were proven. However, in addition to security requirements, the lightweight nature of the proposed scheme is necessary for a resource-constrained IoV. Otherwise, it will be difficult to apply to actual IoV environments. Therefore, we designed simulation experiments based on common methods to analyze the communication protocols for the IoV, which start from the computation and communication overheads. The computation overhead mainly

involves the computation time of the equations in the signcryption and unsigncryption stage, whereas the communication overhead involves the bandwidth requirement for ciphertext transmission.

Specifically, the computation and communication costs of our LS-MRCLSC scheme are compared to those of the schemes in [7–11]. We utilize the JPBC library [48] to simulate cryptographic operations on Orange Pi Zero 2 with a 1.5 GHz quad-core ARM Cortex-A53 CPU and 1 GB DDR3 of RAM. The Orange Pi Zero 2 is shown in Figure 3. Figure 4 shows the implementation. Without loss of generality, we choose the Type A elliptic curve, whose parameters are shown in Table 3. For convenience, we presume the number of receivers n is 100.

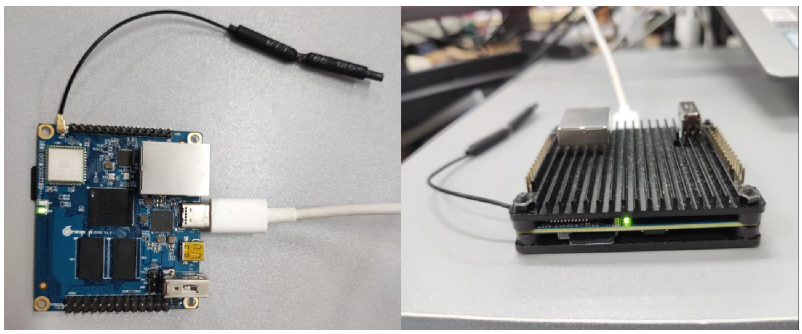

Figure 3. Orange Pi Zero 2.

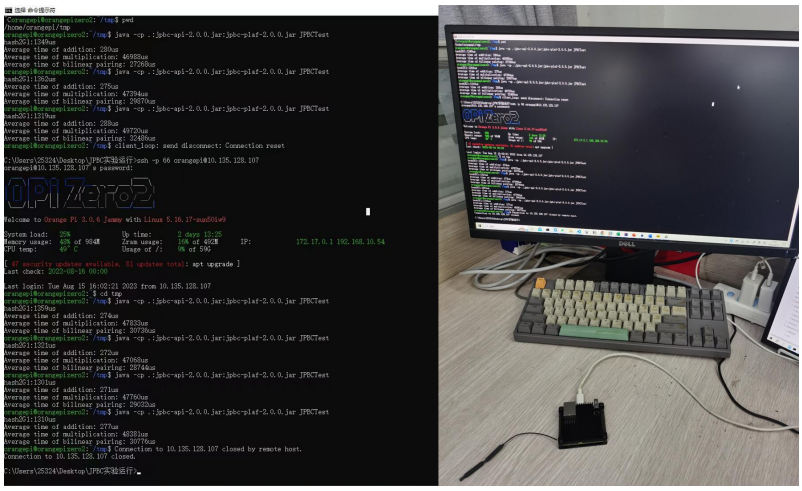

Figure 4. The implementation of the proposed LS-MRCLSC scheme.

Table 3. Elliptic curve parameters.

Item	Parameter
Elliptic curve equation	$y^2 = x^3 + x$
Order of group $\mathbb{G}$	512 bits
Order of $\mathbb{Z}_q^*$	160 bits

6.1. Computation Cost

We mainly compare the computation costs of the signcryption and unsigncryption algorithms. As the schemes in [7–11], as well as our LS-MRCLSC scheme, are all based on ECC, we only consider the operation times of scalar multiplication T_{sm}, point addition T_{pa}, and map-to-point hash T_h. Specifically, the general hash function operation time, the computation time of F_{index}, and the modular operation time are negligible. Table 4 shows the runtimes of the cryptographic operations.

Table 4. Runtimes of cryptographic operations.

Operation	Abbreviation	Runtime (ms)
Scalar multiplication	T_{sm}	11.63
Point addition	T_{pa}	0.059
Map-to-point hash	T_h	25.869

The comparison results of the computation times are shown in Table 5.

Table 5. Comparison of computation costs.

Scheme	Signcryption	Unsigncryption	Total Cost
Scheme [7]	$(2n+1)T_{sm}+2nT_{pa}$	$3T_{sm}+2T_{pa}$	$(2n+4)T_{sm}+(2n+2)T_{pa}$
Scheme [8]	$(3n+1)T_{sm}+2nT_{pa}+T_h$	$5T_{sm}+4T_{pa}+T_h$	$(3n+6)T_{sm}+(2n+4)T_{pa}+2T_h$
Scheme [9]	$(2n+1)T_{sm}+nT_{pa}$	$4T_{sm}+2T_{pa}$	$(2n+5)T_{sm}+(n+2)T_{pa}$
Scheme [10]	$(2n+1)T_{sm}+2nT_{pa}$	$5T_{sm}+3T_{pa}$	$(2n+6)T_{sm}+(2n+3)T_{pa}$
Scheme [11]	$(2n+1)T_{sm}+2nT_{pa}$	$5T_{sm}+3T_{pa}$	$(2n+6)T_{sm}+(2n+3)T_{pa}$
Ours	$(n+1)T_{sm}$	$4T_{sm}+3T_{pa}$	$(n+5)T_{sm}+3T_{pa}$

In our LS-MRCLSC scheme, sender V_i computes $Z=zP$, $U_{R_i}=zX_{R_i}(i=1,2,\ldots,n)$. Thus, sender V_i needs to execute $(n+1)T_{sm}$, which costs 1174.63 ms. In the message unsigncryption stage, receiver V_{R_i} computes $U'_{R_i}=x_{R_i}Z_i$ to obtain the decryption key and checks whether $sig_{R_i}P=Z_i+h_4^{R_i}(X_i+Y_i+h_{3i}P_{pub})$ holds. Therefore, receiver V_{R_i} needs to execute $4T_{sm}+3T_{pa}$, and the time consumed is 46.70 ms. Therefore, the total computation cost is 1221.33 ms.

Similarly, in the message signcryption stage, the schemes in [7–11] consume 2349.43 ms, 3538.30 ms, 2343.53 ms, 2349.43 ms, and 2349.43 ms, respectively. In the message unsigncryption stage, the computation cost is 35.01 ms, 84.26 ms, 46.64 ms, 58.33 ms, and 58.33 ms, and the total cost is 2384.44 ms, 3622.55 ms, 2390.17 ms, 2407.76 ms, and 2407.76 ms, respectively.

According to Figure 5, we can see that compared to the schemes in [7–11], the computation cost of signcryption in our scheme was reduced by 50.00%, 66.80%, 49.87%, 50.00%, and 50.00%, respectively.

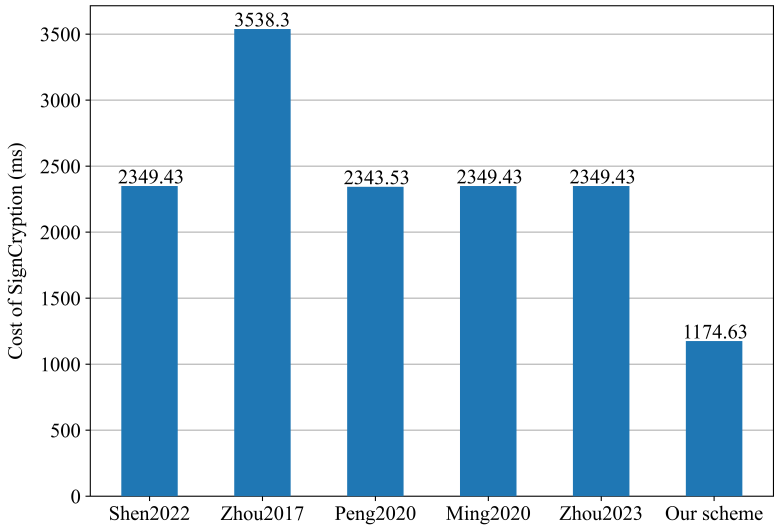

Figure 5. Computation costs of signcryption compared to schemes in [7–11].

As is shown in Figure 6, the computation cost of unsigncryption in our scheme was essentially equal to that in [9] and was reduced by 44.54%, 19.94%, and 19.94% compared

to the schemes in [8,10,11], respectively. Although our cost was slightly higher than that in [7], that scheme could not achieve unlinkability nor resist KGC damage attacks.

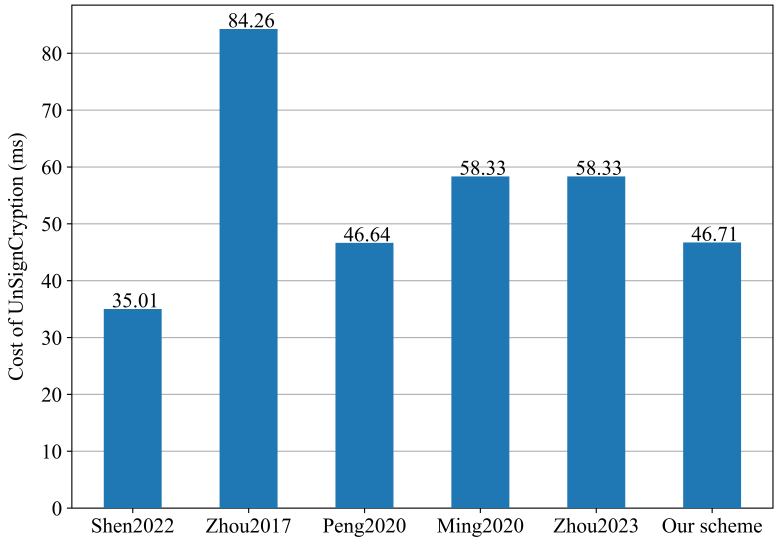

Figure 6. Computation costs of unsigncryption compared to schemes in [7–11].

Figure 7 illustrates the total time cost of the sender. When n = 100, the cost of our scheme was reduced by 48.77%, 66.28%, 48.90%, 49.27%, and 49.27% compared to the schemes in [7–11], respectively. With an increasing number of receivers, our scheme appears to be more advantageous. Hence, our scheme is more efficient and practical.

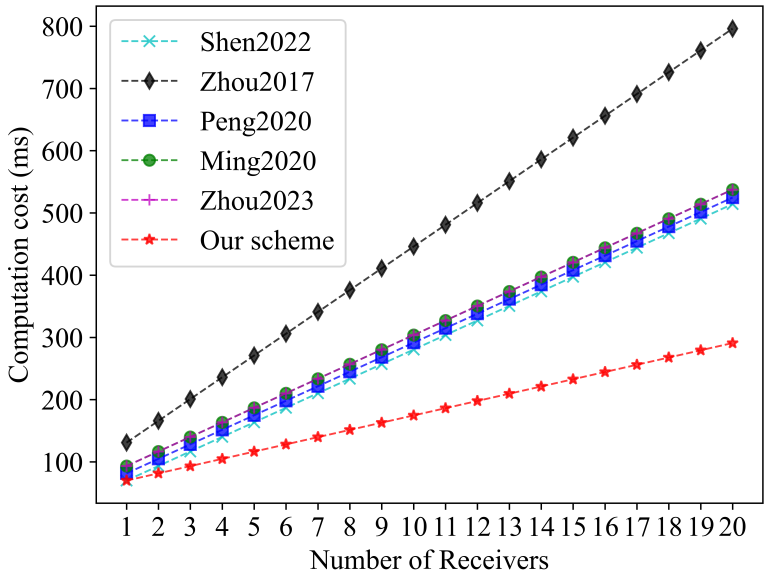

Figure 7. Total time cost of the sender with increasing receivers compared to schemes in [7–11].

6.2. Communication Cost

Table 6 presents the size of an element in groups $\mathbb{G}$ and $\mathbb{Z}_q^*$. Moreover, we neglect the overhead of the timestamps and the encrypted message $|m|$ in all schemes.

Table 6. Notations and lengths.

Notation	Description	Length (bits)
$\|\mathbb{G}\|$	The length of an element in $\mathbb{G}$	1024
$\|\mathbb{Z}_q^*\|$	The length of an element in $\mathbb{Z}_q^*$	160

For 100 receivers, the transmitted data of our scheme $C_m = \{Z, T, CT\}$ are composed of Z, $T = \{t_{R_1,1}, t_{R_2,1}, \ldots, t_{R_n,1}\}$, and $CT = \{c_{R_1}, c_{R_2}, \ldots, c_{R_3}\}$.

The length of C_m is $n|\mathbb{Z}_q^*| + |\mathbb{G}|$ =17,024 bits. Similarly, we can compute the computation overheads of the schemes in [7–11]. A comparison of the ciphertext lengths is shown in Table 7.

Table 7. Comparison of communication costs.

Scheme	Ciphertext Length (bits)
Scheme [7]	$(n+2)\|\mathbb{Z}_q^*\| + 2\|\mathbb{G}\| =$ 18,368
Scheme [8]	$\|\mathbb{Z}_q^*\| + (n+1)\|\mathbb{G}\| =$ 103,584
Scheme [9]	$(2n+1)\|\mathbb{Z}_q^*\| =$ 32,160
Scheme [10]	$(n+1)\|\mathbb{Z}_q^*\| + \|\mathbb{G}\| =$ 17,184
Scheme [11]	$(n+1)\|\mathbb{Z}_q^*\| + \|\mathbb{G}\| =$ 17,184
Ours	$n\|\mathbb{Z}_q^*\| + \|\mathbb{G}\| =$ 17,024

According to Figure 8, the communication overhead of the LS-MRCLSC scheme was reduced by 7.32%, 83.57%, 47.06%, 0.93%, and 0.93% compared to the schemes in [7–11], respectively.

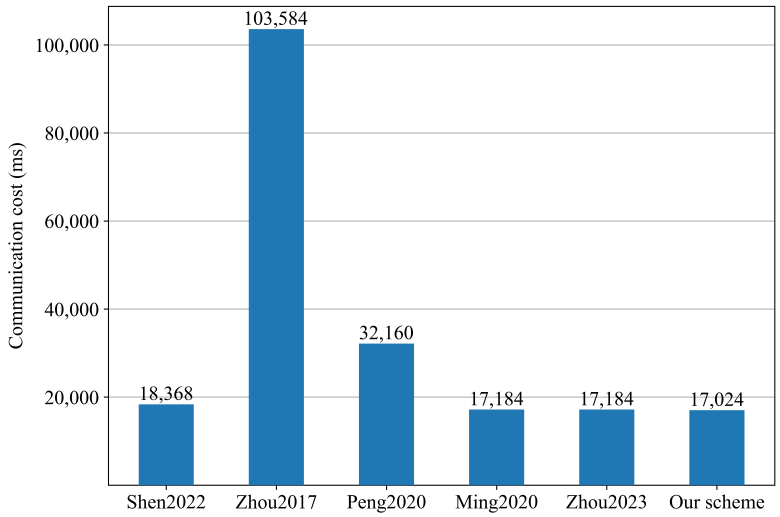

Figure 8. Communication costs compared to schemes in [7–11].

Based on Figure 9, we can see that the communication cost of the sender in the LS-MRCLSC scheme was slightly higher than that in [9] at the beginning, whereas when the number of receivers increased (about $n \geq 6$), our scheme performed much better. Moreover, our scheme fulfilled more security requirements. Consequently, our scheme had a lower communication overhead compared to the schemes in [7–11] when applied to multi-receiver data transmission scenarios.

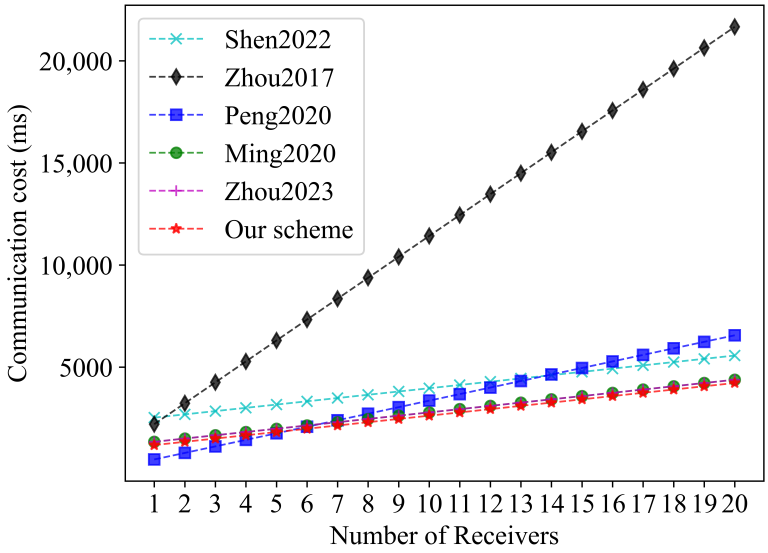

Figure 9. Communication cost of the sender with increasing receivers compared to schemes in [7–11].

6.3. Discussion

1. **The feasibility and scalability of the proposed LS-MRCLSC scheme.**
   - **Lightweight and feasible.** The proposed LS-MRCLSC adopts multi-cast communication to reduce communication time and improve driving efficiency. It broadcasts traffic-related messages to neighboring vehicles in as short a time as possible. In addition, considering the limited computation and storage resources of OBUs and RSUs in the IoV, we have simulated the computation and communication costs using Orange Pi (with fewer resources compared to actual OBUs). The experimental results are more comparable to similar schemes. Therefore, the LS-MRCLSC can be easily applied to the resource-constrained IoV.
   - **Practical and scalable.** The proposed LS-MRCLSC employs multiple KGCs instead of the traditional single KGC (the security assumption is too strong and is prone to SPoFs), which aligns more closely with the needs of practical applications. Moreover, multiple KGCs are independently distributed across different sites. In the system initialization stage, only one round of online interaction is needed for the KGCs. They generate their own sub-keys using the FVSS algorithm and there is no need for mutual trust. After generating the public key P_{pub} for verification, the SMK can be deleted. In this way, the maintenance and management of the SMK can be avoided. When the vehicle initiates a PPK request, each KGC generates part of the PPK independently, and the vehicle computes the complete PPK upon receiving t shares. Hence, the LS-MRCLSC is practical and scalable in an actual IoV environment.
2. **Compression algorithms in the IoV.** Compression and decompression algorithms [49,50] are usually a set of deterministic algorithms and are publicly available. Data with low-security requirements can be transmitted directly or after compression, which greatly reduces the communication bandwidth requirements. However, when transmitting data with strict privacy-preserving demands, the compression algorithms are unsuitable. Once an attacker intercepts a piece of compressed data, it can directly use a decompression algorithm to decompress and obtain the plaintext. In other words, compression algorithms cannot achieve data confidentiality, whereas our proposed LS-MRCLSC scheme has proven to be secure. Of course, encrypting the compressed data is a credible approach with security considerations. But in this case, the computational overhead would be very high, which is intolerable for the resource-constrained IoV environment. All in all, we usually require the algorithm to be public and the key

to be secret. If the algorithm is kept secret, once the attacker breaks the algorithm, the consequences will be very serious. As for key secrecy, we can ensure the security of the algorithm by updating the key periodically, which is more practical.

3. **Digital twin technology in the IoV.** Digital twin technology [51–53] is a virtual counterpart to actual physical devices (entities). It can enhance the security and efficiency of the IoV ecosystem, particularly in terms of vehicle data monitoring, predictive maintenance, and anomaly detection. For instance, digital twin-based penetration tests could enable relevant tests virtually (instead of on a real system) during both the operation phase and the engineering phase to fix vulnerabilities early in the lifecycles of cyber-physical systems (CPSs) [54]. Thus, the integration of digital twins within the proposed LS-MRCLSC scheme may be a research direction to consider.

7. Conclusions

In this paper, we propose an LS-MRCLSC scheme for the IoV without secure channels. The leveraging of an MMSC structure enables vehicles to transmit a batch of messages to the designated receivers in one report. In addition, multiple KGCs are employed to resist KGC damage attacks and avoid SPoFs. Moreover, we have proven that the LS-MRCLSC scheme satisfies data confidentiality and message unforgeability under the ROM. Security proofs and performance evaluations show that the LS-MRCLSC scheme can provide vehicles with secure communication and privacy protection at a lower cost in contrast to related schemes.

Public key replacement attacks are common in many CPPA protocols. Therefore, in future work, we will try to utilize a blockchain to design a secure CPPA scheme for the IoV that can withstand such attacks. The key materials of vehicles can be stored on the blockchain for public key queries with pseudonyms. The tamper-proof property of the blockchain will eliminate public key replacement attacks using effective and verifiable approaches. Meanwhile, the communication costs of vehicles' public keys will be saved, which is another advantage for resource-limited vehicles.

Author Contributions: Conceptualization, G.X. and X.Y.; methodology, G.X.; software, G.X.; validation, X.Y. and X.L.; formal analysis, G.X. and X.L.; investigation, G.X.; resources, G.X.; data curation, X.Y.; writing—original draft preparation, G.X.; writing—review and editing, X.Y. and X.L.; visualization, G.X.; supervision, X.Y. and X.L.; project administration, X.Y.; funding acquisition, X.Y. and X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the Henan Key Laboratory of Network Cryptography Technology under grant No. LNCT2022-A17, and in part by the Postgraduate Research and Practice Innovation Program of Jiangsu Province under grant No. KYCX22_3501 .

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are unavailable due to privacy.

Acknowledgments: We sincerely thank the reviewers and the editor for their review and approval.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

IoV	Internet of Vehicles
LS-MRCLSC	Lightweight and Secure Multi-Message Multi-Receiver Certificateless Signcryption
ECC	Elliptic Curve Cryptography
KGC	Key Generation Center
SMK	System Master Key

FVSS	Feldman's Verifiable Secret Sharing
SPoF	Single Point of Failure
ROM	Random Oracle Model
DSRC	Dedicated Short-Range Communications
C-V2X	Cellular Vehicle-to-Everything
TMA	Traffic Management Authority
PPK	Partial Private Key
DoS	Denial of Service
APT	Advanced Persistent Threat
TSS	Threshold Secret Sharing
OBU	On-Board Unit
RSU	Road-Side Unit
V2V	Vehicle-to-Vehicle
MMSC	Multi-Message Multi-Receiver Signcryption
DDoS	Distributed Denial of Service
PoW	Proof-of-Work
PoT	Proof-of-Trajectory
CPPA	Conditional Privacy-Preserving Authentication
PKI	Public Key Infrastructure
MRCLSC	Multi-Receiver Certificateless Signcryption
IoT	Internet of Things
VANET	Vehicular Ad Hoc Network
ECCDHP	Elliptic Curve Computational Diffie–Hellman Problem
PPT	Probabilistic Polynomial Time
ECDLP	Elliptic Curve Discrete Logarithm Problem
IND-CCA2	Indistinguishability against a Chosen Ciphertext Attack Adaptively
EUF-CMA	Existential Unforgeability under a Chosen Message Attack
TPD	Tamper-Proof Device
CPS	Cyber-Physical System

References

1. Yang, Y.; Zhang, L.; Zhao, Y.; Choo, K.K.R.; Zhang, Y. Privacy-Preserving Aggregation-Authentication Scheme for Safety Warning System in Fog-Cloud Based VANET. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 317–331. [CrossRef]
2. Yang, X.; Li, X.; Li, T.; Wang, X.; Wang, C.; Li, B. Efficient and anonymous multi-message and multi-receiver electronic health records sharing scheme without secure channel based on blockchain. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4371. [CrossRef]
3. Wang, Y.; Liu, Y.; Tian, Y. ISC-CPPA:Improverd-Security Certificateless Conditional Privacy-Preserving Authentication Scheme With Revocation. *IEEE Trans. Veh. Technol.* **2022**, *71*, 12304–12314. [CrossRef]
4. Shamir, A. How to Share a Secret. *Commun. ACM* **1979**, *22*, 612–613. [CrossRef]
5. Mundhe, P.; Verma, S.; Venkatesan, S. A comprehensive survey on authentication and privacy-preserving schemes in VANETs. *Comput. Sci. Rev.* **2021**, *41*, 100411. [CrossRef]
6. Chor, B.; Goldwasser, S.; Micali, S.; Awerbuch, B. Verifiable secret sharing and achieving simultaneity in the presence of faults. In Proceedings of the 26th Annual Symposium on Foundations of Computer Science (Sfcs 1985), Washington, DC, USA, 21–23 October 1985; pp. 383–395.
7. Shen, J.; Gui, Z.; Chen, X.; Zhang, J.; Xiang, Y. Lightweight and Certificateless Multi-Receiver Secure Data Transmission Protocol for Wireless Body Area Networks. *IEEE Trans. Dependable Secur. Comput.* **2022**, *19*, 1464–1475. [CrossRef]
8. Zhou, Y.; Yang, B.; Zhang, W. Multi-receiver and multi-message of certificateless signcryption scheme. *Chin. J. Comput.* **2017**, *40*, 1714–1724.
9. Peng, C.; Chen, J.; Obaidat, M.S.; Vijayakumar, P.; He, D. Efficient and Provably Secure Multireceiver Signcryption Scheme for Multicast Communication in Edge Computing. *IEEE Internet Things J.* **2020**, *7*, 6056–6068. [CrossRef]
10. Ming, Y.; Yu, X.; Shen, X. Efficient Anonymous Certificate-Based Multi- Message and Multi-Receiver Signcryption Scheme for Healthcare Internet of Things. *IEEE Access* **2020**, *8*, 153561–153576. [CrossRef]
11. Zhou, Y.; Xu, R.; Qiao, Z.; Yang, B.; Xia, Z.; Zhang, M. An Anonymous and Efficient Multi-Message and Multi-Receiver Certificateless Signcryption Scheme for VANET. *IEEE Internet Things J.* **2023**. [CrossRef]
12. Cui, J.; Wu, D.; Zhang, J.; Xu, Y.; Zhong, H. An efficient authentication scheme based on semi-trusted authority in VANETs. *IEEE Trans. Veh. Technol.* **2019**, *68*, 2972–2986. [CrossRef]
13. Gao, Y.; Wu, H.; Song, B.; Jin, Y.; Luo, X.; Zeng, X. A distributed network intrusion detection system for distributed denial of service attacks in Vehicular Ad Hoc Network. *IEEE Access* **2019**, *7*, 154560–154571. [CrossRef]

14. Baza, M.; Nabil, M.; Mahmoud, M.M.; Bewermeier, N.; Fidan, K.; Alasmary, W.; Abdallah, M. Detecting sybil attacks using proofs of work and location in VANETs. *IEEE Trans. Dependable Secur. Comput.* **2020**, *19*, 39–53. [CrossRef]
15. Ren, J.; Cheng, Y.; Xu, S. EDPPA: An efficient distance-based privacy preserving authentication protocol in VANETs. *Peer Peer Netw. Appl.* **2022**, *15*, 1385–1397. [CrossRef]
16. Bao, Y.; Qiu, W.; Cheng, X.; Sun, J. Fine-Grained Data Sharing With Enhanced Privacy Protection and Dynamic Users Group Service for the IoV. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 13035–13049. [CrossRef]
17. Lin, C.; He, D.; Huang, X.; Kumar, N.; Choo, K.K.R. BCPPA: A Blockchain-Based Conditional Privacy-Preserving Authentication Protocol for Vehicular Ad Hoc Networks. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 7408–7420. [CrossRef]
18. Saqib, N.U.; Malik, S.U.R.; Anjum, A.; Syed, M.H.; Moqurrab, S.A.; Srivastava, G.; Lin, J.C.W. Preserving Privacy in the Internet of Vehicles (IoV): A Novel Group Leader-based Shadowing Scheme using Blockchain. *IEEE Internet Things J.* **2023**, 1–10. [CrossRef]
19. Tu, S.; Yu, H.; Badshah, A.; Waqas, M.; Halim, Z.; Ahmad, I. Secure Internet of Vehicles (IoV) With Decentralized Consensus Blockchain Mechanism. *IEEE Trans. Veh. Technol.* **2023**, *72*, 11227–11236. [CrossRef]
20. Verma, R. An Efficient Secure VANET Communication Using Multi Authenticate Homomorphic Signature Algorithm. In Proceedings of the 2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Ballar, India, 29–30 April 2023; pp. 1–5.
21. Zhuang, L.; Guo, N.; Chen, Y. TriNymAuth: Triple Pseudonym Authentication Scheme for VANETs Based on Cuckoo Filter and Paillier Homomorphic Encryption. *Sensors* **2023**, *23*, 1164. [CrossRef]
22. Al-Riyami, S.S.; Paterson, K.G. Certificateless public key cryptography. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, Warsaw, Poland, 4–8 May 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp.452–473.
23. Ellison, C.; Schneier, B. Ten risks of PKI: What you're not being told about public key infrastructure. *Comput. Secur. J.* **2000**, *16*, 1–7.
24. Shamir, A. Identity-based cryptosystems and signature schemes. In Proceedings of the Advances in Cryptology: Proceedings of CRYPTO 84 4, Paris, France, 7–11 April 1985; Springer: Berlin/Heidelberg, Germany, 1985; pp. 47–53.
25. Horng, S.J.; Tzeng, S.F.; Huang, P.H.; Wang, X.; Li, T.; Khan, M.K. An efficient certificateless aggregate signature with conditional privacy-preserving for vehicular sensor networks. *Inf. Sci.* **2015**, *317*, 48–66. [CrossRef]
26. Ali, I.; Chen, Y.; Ullah, N.; Kumar, R.; He, W. An Efficient and Provably Secure ECC-Based Conditional Privacy-Preserving Authentication for Vehicle-to-Vehicle Communication in VANETs. *IEEE Trans. Veh. Technol.* **2021**, *70*, 1278–1291. [CrossRef]
27. Shim, K.A. Comments on "Revocable and Scalable Certificateless Remote Authentication Protocol With Anonymity for Wireless Body Area Networks". *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 81–82. [CrossRef]
28. Wu, C.; Huang, H.; Zhou, K.; Xu, C. Cryptanalysis and improvement of a new certificateless signature scheme in the standard model. *China Commun.* **2021**, *18*, 151–160. [CrossRef]
29. Zheng, Y. Digital signcryption or how to achieve cost(signature & encryption) $\ll$ cost(signature) + cost(encryption). In Proceedings of the Advances in Cryptology—CRYPTO'97: 17th Annual International Cryptology Conference, Santa Barbara, CA, USA, 17–21 August 1997; Proceedings 17; Springer: Berlin/Heidelberg, Germany, 1997; pp. 165–179.
30. Selvi, D.S.; Vivek, S.S.; Shukla, D.; Rangan, C. Efficient and Provably Secure Certificateless Multi-receiver Signcryption. In Proceedings of the Provable Security: Second International Conference, ProvSec 2008, Shanghai, China, 30 October–1 November 2008; Proceedings 2; Volume 5324, pp. 52–67.
31. Miao, S.; Zhang, F.; Zhang, L. Cryptanalysis of a certificateless multi-receiver signcryption scheme. In Proceedings of the 2010 International Conference on Multimedia Information Networking and Security, Nanjing, China, 4–6 November 2010; pp. 593–597.
32. Niu, S.; Li, Z.; Wang, C. Privacy-preserving multi-party aggregate signcryption for heterogeneous systems. In Proceedings of the Cloud Computing and Security: Third International Conference, ICCCS 2017, Nanjing, China, 16–18 June 2017; Revised Selected Papers, Part II 3; Springer: Berlin/Heidelberg, Germany, 2017; pp. 216–229.
33. Li, H.; Pang, L. Cryptanalysis of Wang et al.'s improved anonymous multi-receiver Identity-Based encryption scheme. *IET Inf. Secur.* **2014**, *8*, 8–11. [CrossRef]
34. Pang, L.; Kou, M.; Wei, M.; Li, H. Efficient Anonymous Certificateless Multi-Receiver Signcryption Scheme Without Bilinear Pairings. *IEEE Access* **2018**, *6*, 78123–78135. [CrossRef]
35. Yu, X.; Zhao, W.; Tang, D. Efficient and provably secure multi-receiver signcryption scheme using implicit certificate in edge computing. *J. Syst. Archit.* **2022**, *126*, 102457. [CrossRef]
36. Gao, R.; Zeng, J.; Deng, L. Efficient Certificateless Anonymous Multi-Receiver Encryption Scheme without Bilinear Parings. *Math. Probl. Eng.* **2018**, *2018*, 1486437. [CrossRef]
37. Chen, L.; Li, J.; Zhang, Y. Anonymous Certificate-Based Broadcast Encryption With Personalized Messages. *IEEE Trans. Broadcast.* **2020**, *66*, 867–881. [CrossRef]
38. Lu, Y.; Li, J.; Zhang, Y. Privacy-Preserving and Pairing-Free Multirecipient Certificateless Encryption With Keyword Search for Cloud-Assisted IIoT. *IEEE Internet Things J.* **2020**, *7*, 2553–2562. [CrossRef]
39. Seo, M.; Kim, K. Electronic Funds Transfer Protocol Using Domain-Verifiable Signcryption Scheme. *Lect. Notes Comput. Sci.* **2000**, *1787*, 269–277.

40. Elkamchouchi, D.H. A chaotic public key multi-message multi-recipients signcryption scheme (CPK-MM-MR-SS). In Proceedings of the 14th International Conference on Information Systems Security (ICISS 2008), Bangalore, India, 17–19 December 2008; pp. 30–34.
41. Elkamchouchi, H.; Hagras, E. An efficient Public Key Multi-Messages Multi-Recipients Elliptic Curve Signcryption (PK-MM-ECS) scheme. In Proceedings of the 2008 National Radio Science Conference, Tanta, Egypt, 18–20 March 2008; pp. 1–10.
42. Pang, L.; Wei, M.; Li, H. Efficient and Anonymous Certificateless Multi-Message and Multi-Receiver Signcryption Scheme Based on ECC. *IEEE Access* **2019**, *7*, 24511–24526. [CrossRef]
43. Nizamuddin.; Umar, A.I.; Waheed, A.; ul Amin, N. An Efficient Multi-Message Multi-Receiver Signcryption Scheme with Forward Secrecy on Elliptic Curves. *Cryptol. ePrint Arch.* **2015**, *2015*, 655.
44. Wang, C.; Liu, C.; Li, Y.; Qiao, H.; Chen, L. Multi-message and multi-receiver heterogeneous signcryption scheme for Ad-Hoc Network. *Inf. Secur. J. Glob. Perspect.* **2017**, *26*, 136–152. [CrossRef]
45. ur Rahman, A.; Ullah, I.; Naeem, M.; Anwar, R.; Khattak, H.; Ullah, S. A lightweight multi-message and multi-receiver heterogeneous hybrid signcryption scheme based on hyper elliptic curve. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*. [CrossRef]
46. Qiu, J.; Fan, K.; Zhang, K.; Pan, Q.; Li, H.; Yang, Y. An Efficient Multi-Message and Multi-Receiver Signcryption Scheme for Heterogeneous Smart Mobile IoT. *IEEE Access* **2019**, *7*, 180205–180217. [CrossRef]
47. Hellman, M. New directions in cryptography. *IEEE Trans. Inf. Theory* **1976**, *22*, 644–654.
48. Angelo, D.C.; Vincenzo, I. jPBC: Java pairing based cryptography. In Proceedings of the 16th IEEE Symposium on Computers and Communications, (ISCC 2011) , Kerkyra, Corfu, Greece, 28 June–1 July 2011; pp. 850–855.
49. Wiseman, Y. Adapting the H. 264 Standard to the Internet of Vehicles. *Technologies* **2023**, *11*, 103. [CrossRef]
50. Rakhmanov, A.; Wiseman, Y. Compression of GNSS Data with the Aim of Speeding up Communication to Autonomous Vehicles. *Remote Sens.* **2023**, *15*, 2165. [CrossRef]
51. Piromalis, D.; Kantaros, A. Digital twins in the automotive industry: The road toward physical-digital convergence. *App. Syst. Innov.* **2022**, *5*, 65. [CrossRef]
52. Tsaramirsis, G.; Kantaros, A.; Al-Darraji, I.; Piromalis, D.; Apostolopoulos, C.; Pavlopoulou, A.; Alrammal, M.; Ismail, Z.; Buhari, S.M.; Stojmenovic, M.; et al. A modern approach towards an industry 4.0 model: From driving technologies to management. *J. Sens.* **2022**, *2022*, 5023011. [CrossRef]
53. Kantaros, A.; Piromalis, D.; Tsaramirsis, G.; Papageorgas, P.; Tamimi, H. 3D printing and implementation of digital twins: Current trends and limitations. *App. Syst. Innov.* **2021**, *5*, 7. [CrossRef]
54. Veledar, O.; Damjanovic-Behrendt, V.; Macher, G. Digital twins for dependability improvement of autonomous driving. In Proceedings of the European Conference on Software Process Improvement, Edinburgh, UK, 18–20 September 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 415–426.