electronics

MDPI

Article

Unlinkable and Revocable Signcryption Scheme for VANETs

Lihui Li [1,2], Dongmei Chen [1], Yining Liu [3], Yangfan Liang [4,5,6], Yujue Wang [7,*] and Xianglin Wu [8,9]

[1] School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China; llhljz@mails.guet.edu.cn (L.L.); candy.chen.potato@mails.guet.edu.cn (D.C.)
[2] Guangxi Universities Key Laboratory of Application Technology of Intelligent Connected Vehicle, Guangxi Vocational Normal University, Nanning 530007, China
[3] School of Data Science and Artificial Intelligence, Wenzhou University of Technology, Wenzhou 325027, China; ynliu@guet.edu.cn
[4] Provincial Key Laboratory of Multimodal Perceiving and Intelligent Systems, Jiaxing University, Jiaxing 314001, China; 19032205006@mails.guet.edu.cn
[5] Key Laboratory of Medical Electronics and Digital Health of Zhejiang Province, Jiaxing University, Jiaxing 314001, China
[6] Engineering Research Center of Intelligent Human Health Situation Awareness of Zhejiang Province, Jiaxing University, Jiaxing 314001, China
[7] Hangzhou Innovation Institute, Beihang University, Hangzhou 310056, China
[8] Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China; gtwxl@mails.guet.edu.cn
[9] School of Artificial Intelligence, Hezhou University, Hezhou 542899, China
* Correspondence: wyujue@buaa.edu.cn

Abstract: Vehicular ad-hoc networks (VANETs) can significantly improve the level of urban traffic management. However, the sender unlinkability has become an intricate issue in the field of VANETs' encryption. As the sender signcrypts a message, the receiver has to use the sender's identity or public key to decrypt it. Consequently, the sender can be traced using the same identity or public key, which poses some security risks to the sender. To address this issue, we present an unlinkable and revocable signcryption scheme (URSCS), where an efficient and powerful signcryption mechanism is adopted for communication. The sender constructs a polynomial to generate a unique session key for each communication, which is then transmitted to a group of receivers, enabling the same secret message to be sent to multiple receivers. Each time a secret message is sent, a new key pair is generated, and an anonymization mechanism is introduced to conceal the true identity of the vehicle, thus preventing malicious attackers from tracing the sender through the public key or the real identity. With the introduction of the identification public key, this scheme supports either multiple receivers or a single receiver, where the receiver can be either road side units (RSUs) or vehicles. Additionally, a complete revocation mechanism is constructed with extremely low communication overhead, utilizing the Chinese remainder theorem (CRT). Formal and informal security analyses demonstrate that our URSCS scheme meets the expected security and privacy requirements of VANETs. The performance analysis shows that our URSCS scheme outperforms other represented schemes.

Keywords: VANETs; unlinkable; signcryption; multi-receiver; conditional anonymous; Chinese remainder theorem

Citation: Li, L.; Chen, D.; Liu, Y.; Liang, Y.; Wang, Y.; Wu, X. Unlinkable and Revocable Signcryption Scheme for VANETs. *Electronics* **2024**, *13*, 3164. https://doi.org/10.3390/electronics13163164

Academic Editor: Aryya Gangopadhyay

Received: 17 June 2024
Revised: 24 July 2024
Accepted: 3 August 2024
Published: 10 August 2024

1. Introduction

With the increasing popularity of cars in ordinary families and the increasing complexity of urban roads, traffic accidents have become a major risk factor in modern society [1]. In order to reduce the traffic dangers encountered by people in daily life and improve the efficiency of information sharing between vehicles, VANETs have emerged [2]. Vehicles move at high speeds, and VANETs mainly use the dedicated short-range communications (DSRC) protocol [3], so it is easy for malicious attackers to listen, intercept, modify, and retransmit messages [4], which poses great security risks to VANETs. Therefore, ensuring the

legitimacy of the information source, ensuring that the received message is not tampered with, and ensuring the secret transmission of the message have become basic requirements in the field of vehicle networking security [5–7].

In recent years, signcryption technology has proven capable of completing the above three required functions in one operation simultaneously, which greatly reduces communication and computation overhead [8]. It has been widely used, especially in multi-receiver scenarios, such as VANETs, mobile communication devices, wireless body area networks, and maritime transportation systems. However, the traditional multi-receiver signcryption technology requires the sender to negotiate with each receiver and send different ciphertext, which significantly reduces the efficiency of message transmission [9–17]. Additionally, there is a problem that can be traced back to the sender. When the same sender sends multiple messages, it needs to use its own private key for signcryption, and the receiver uses the public key or real identity of the sender for unsigncryption. Therefore, malicious attackers can trace the sender based on the same public key or real identity used in the process of multiple signcryption. In order to solve this problem, Liang et al. [18] first proposed a scheme with multiple receivers and unlinkable to senders. An anonymous transmission is taken each time a message is sent and a key pair is updated by the sender, and uses CRT to transmit the public session key. However, their scheme can only be used for one-to-many transmission for road side units (RSUs).

In the context of VANETs, malicious vehicles will send malicious messages for various reasons, leading to various traffic problems. Therefore, TCC's ability to track and revoke malicious vehicles is particularly important [19]. In VANETs, for the implementation of the revocation mechanism, the schemes [20–22] allocate a revocation list to each vehicle, resulting in high storage overhead for the entire system. At the same time, TCC needs to frequently update the revocation list to all vehicles, which also incurs high communication overhead. Scheme [18] implements revocation mechanisms by constructing polynomials, but since the number of parameters in the publicly disclosed revocation public key equals the number of vehicles, this method consequently faces the problem of high transmission overhead. In a resource-constrained VANET environment, a revocation mechanism can only be practically feasible if its cost is sufficiently low.

With the motivation of addressing the aforementioned issues, this paper proposes an unlinkable and revocable signcryption scheme (URSCS). Our main contributions are listed as follows:

- For the first time, in the scenario of vehicle networking, our URSCS scheme truly realizes the unlinkable communication for the sender in one-to-many or one-to-one modes at the same time. By introducing the identification of public and private key pairs instead of real identities, the receivers can be either the vehicles or the RSUs. In order to achieve unlinkability of the sender, our URSCS scheme takes the following approach: First, we provide conditional privacy through a pseudonym mechanism, which effectively hides the true identity of the vehicle, so that malicious attackers cannot track the sender through the real identity. Second, the vehicle updates its key pair and pseudonym before each signcryption, ensuring that a malicious attacker cannot link the sender to the message based on the public keys or pseudonyms;
- Based on CRT, we propose a comprehensive revocation key update mechanism that incurs low communication costs. Regardless of the number of vehicles within the domain covered by a Traffic Control Center (TCC), the revocation key information transmitted remains at 96 bytes. Consequently, the TCC can efficiently remove malicious vehicles from the system and promptly update the revocation public key to prevent them from transmitting further harmful information. Additionally, by incorporating several time thresholds, our URSCS scheme effectively prevents malicious vehicles from continuing to send messages, even if their revocation keys are reassigned to new vehicles;

- We conduct formal and informal security analyses of our URSCS scheme, which show that our scheme satisfies various security requirements and can effectively resist various known potential attacks;
- Rigorous performance evaluation confirms that with the increase in the number of receiving ends, the communication and computation overhead of the proposed URSCS scheme has significant advantages compared with other related schemes.

The remainder of this paper is organized as follows: the related works are introduced in Section 2. Some preliminaries are briefly introduced in Section 3, including the elliptic curve cryptosystem, classical hard problems, and the basic concepts of CRT. Section 4 outlines the system model and security and privacy goals of our URSCS scheme. The detailed design of our URSCS scheme is described in Section 5. And the security analysis and performance assessment are presented in Sections 6 and 7, respectively. Finally, we conclude this paper in Section 8.

2. Related Work

In this section, we will extensively discuss the security techniques involved in transmitting information over an open channel, mainly focusing on signcryption, revocability, anonymity, and unlinkability.

2.1. Signcryption

Zheng Yuliang [23] first introduced a new cryptographic primitive named signcryption, performing both signature and encryption, and simultaneously ensuring the authentication, integrity, and confidentiality of the message. Compared to the sign-then-encrypt method, this mechanism also reduced the computation overhead and communication overhead, making it suitable for resource-constrained application scenarios [24,25]. In 2003, Al-Riyami et al. proposed a new cryptography primitive called certificateless public key cryptography in [26], which addresses both the certificate management problem in traditional public key infrastructure (PKI) [27] and the key escrow problem in identity-based cryptosystems [28]. Specifically, a certificateless public key cryptography system divides a user's public key and private key into two parts. The key generation center (KGC) generates only part of the public key and private key for the user, while the other part is generated by the user themselves, and the part generated by the user is referred to as the secret value. In 2008, Barbosa Manuel and Farshim Pooya proposed the first certificateless signcryption (CLSC) scheme [29], which combines the certificateless cryptosystem and signcryption. Since then, researchers have proposed a number of CLSC schemes [30–33]. However, these schemes are based on bilinear pairing operations, and due to the high computation cost of bilinear pairing, they are unsuitable for devices with limited computational resources, such as on-board units (OBUs) in vehicles.

In order to accommodate resource-constrained scenarios such as the VANETs, many researchers began designing certificateless signcryption schemes without pairing operations. Cui et al. designed a new certificateless aggregate signcryption scheme [34], successfully eliminating the expensive pairing operation and thereby improving computational efficiency. However, Zhan et al. pointed out that Cui et al.'s protocol is not guaranteed to be unforgeable under the second type of attack. They subsequently designed a new CLSC scheme [35] which addresses the presence of unforgeability in the second type of attack. Since then, many unpaired certificateless signcryption schemes, such as [36–38], have been developed and applied in numerous resource-constrained scenarios. However, these schemes also face certain challenges. For instance, the scheme proposed by Shen et al. [36] does not incorporate a revocation mechanism, and Yu and Ren's scheme [37] has been proven unable to withstand signature forgery attacks and collusion attacks. Recently, a considerable number of signcryption-based schemes have been developed by numerous researchers for multi-receiver scenarios, aiming to boost communication and computation performance [8,11–17]. However, these schemes still suffer from all kinds of shortcomings. Specifically, the schemes [11,14,15], and Ref. [38] lacked sender unlinkability, enabling the

receiver to associate the sender with a fixed public key, and Ref. [16] failed to provide sender anonymity due to the sender communicating openly with their real identity. In addition, a handful of schemes, specifically [8,17], lacked both a tracing and revocation mechanism.

2.2. Revocability

In order to solve the vehicle revocation problem, many researchers have proposed anonymity-based conditional privacy schemes with revocation function [20–22]. However, in these schemes, to ensure anonymity and unlinkability, the vehicle needs to be pre-loaded with a large number of anonymous public keys and certificates. However, one of the challenges with this type of scheme is the large scale of certificate revocation lists (CRL), resulting in increased storage and verification costs. When a vehicle is revoked, its certificate is added to the CRL, further increasing its size and thus exacerbating transmission delays. Therefore, the scale of the CRL and the cost of exhaustive search become important constraints of this revocation mechanism.

Shim et al. proposed a conditional privacy-preserving authentication scheme based on public key cryptography [39], where public identities are used to verify the legitimacy of vehicle identities, and vehicle anonymity is achieved through two trusted agents, namely, the trace authority and the private key generator. However, Wang et al. [40] pointed out that Shim and Kyung-Ah's scheme [39] would incur significant computational and communication costs with an increase in the number of vehicles, and the revocation process could not meet the requirements of V2V communication, and proposed an enhanced security certificateless conditional privacy-preserving authentication scheme with revocation [40]. However, since the vehicle keys are all generated by the trusted authority, there is an issue with key escrow.

2.3. Anonymity and Unlinkability

In VANETs, anonymity means that no one can infer the sender's identity information from the received message. Unlinkability ensures that no one (including the recipients of the messages) can determine whether two transmitted messages come from the same vehicle [18]. Currently, many researchers have proposed numerous authentication schemes for user anonymity [11,35,41–44]. However, most existing schemes in VANETs cannot fully achieve sender unlinkability. Specifically, the receiver must use the sender's identity or public key to complete the verification process. Consequently, the receiver can link different messages to the same sender through the invariant elements, identity, or public key within the messages. For example, in [42], a new certificate-based identity authentication scheme was proposed to enhance the security of vehicular networks. However, when the sender is the same, the signing and verification process always uses the same pseudonym, allowing attackers to link multiple messages to the same sender through the identical pseudonym. Gayathri et al. [43] proposed a pairing-free certificateless authentication scheme to achieve batch verification in VANETs; however, each signature message generated by user ID_i contains an immutable parameter R_i, an attacker can determine whether two messages are from the same sender by checking if the parameters in the signatures are equal. Liang et al. [18] achieved the first truly sender-unlinkable multi-receiver signcryption scheme. However, in Liang's scheme, the real ID of the receiver must be used when transmitting information, which means that the receivers can only be RSUs, and V2V communication cannot be guaranteed.

3. Preliminaries

This section will provide a detailed introduction to the Chinese remainder theorem, the elliptic curve cryptosystem, and classic hardness assumptions.

3.1. Chinese Remainder Theorem (CRT)

The CRT is an important tool in number theory used to solve a class of problems related to systems of modular congruence equations. Consider a modular congruence system as follows.

$$\begin{cases} x \equiv t_1 \pmod{k_1} \\ x \equiv t_2 \pmod{k_2} \\ x \equiv t_3 \pmod{k_3} \\ \quad \vdots \\ x \equiv t_n \pmod{k_n} \end{cases} \tag{1}$$

where $t_1, t_2, \ldots, t_n$ are given n integers, and $k_1, k_2, \ldots, k_n$ are n coprime positive integers. If the congruence system (1) has a solution, then through the CRT, we can obtain a unique solution $x = \sum_{i=1}^{n} t_i K_i K_i^{-1}$, where $K_i = \prod_{j=1, j \neq i}^{n} K_j$, K_i^{-1} is the multiplicative inverse of K_i modulo k_i. In this paper, the CRT is not used to solve for x in the congruence system; rather, its principles are employed to transmit public parameters.

3.2. Elliptic Curve Cryptosystem ECC

Assume F_p is a finite field defined by modulo p, where p is a large prime number. We define an elliptic curve E over F_p, which is in the form of a set of points $\{(x, y) \in (F_p)^2 | y^2 \equiv x^3 + ax + b(mod\ p), 4a^3 + 27b^2 \neq 0(mod\ p)\}$, where a and b are elements of F_p. And, the combination of an infinite point O along with every point on E comprises an additive elliptic curve group G, that possesses a prime order q. This group G satisfies the subsequent properties:

Point addition: For any $P, Q \in G$, the following conditions hold: if $P = Q$, $2P = P + Q$; if $P = -Q$, $O = P + Q$; and if $P \neq Q$, $R = P + Q$, where $-R$ is the intersection of elliptic curve E and the straight line connecting P and Q, R is the inverse of $-R$ on G.

Scale multiplication: Given $P \in G$ and $n \in Z_q^*$, $nP = P + P + \cdots + P$ (n times).

3.3. Hard Problems

Elliptic Curve Discrete Logarithm Problem (ECDLP): Given $P, \alpha P \in G$, it is hard to compute the value of α in a probabilistic polynomial-time (PPT), where $\alpha \in Z_q^*$.

Computational Diffie–Hellman Problem (CDHP): Given $P, \alpha P, \beta P \in G$, it is hard to compute $\alpha\beta P$ in a PPT, where $\alpha, \beta \in Z_q^*$.

4. System Model and Security Requirements

4.1. System Model

As shown in Figure 1, there are mainly three entities in our URSCS scheme: TCC, RSU, and Vehicle. Due to the huge geographical coverage of VANETs, we assume the existence of multiple TCCs. Communication between vehicles, as well as between vehicles and RSUs, is achieved through open dedicated short-range wireless signals, while communication between RSUs and between TCC and RSUs is performed through wired communication.

TCC: It is assumed to be a trusted and secure entity with unlimited computational capability. It is mainly responsible for the initialization of the entire network, network management, registration of vehicles and RSUs, generating partial private keys, pseudonyms, revocation keys for vehicles, and it can track and revoke malicious vehicles.

RSU: Installed on both sides of various roads, it is a semi-honest entity, meaning it faithfully executes various instructions of the system, but it also tends to be curious and collects various information about the system. Its main responsibilities include collecting, decrypting, forwarding, and verifying the encrypted information of surrounding vehicles, as well as forwarding information from the TCC.

Vehicle: Each vehicle is eligible to be included in the system only after registering with the TCC. Furthermore, each vehicle must be equipped with a tamper-proof device (TPD) and an OBU to securely store and process sensitive data. Vehicles collect real-time traffic

data, signciphers the information, and transmit the generated ciphertext to designated entity sets. At the same time, vehicles are always ready to receive instructions from the TCC and RSUs and respond promptly.

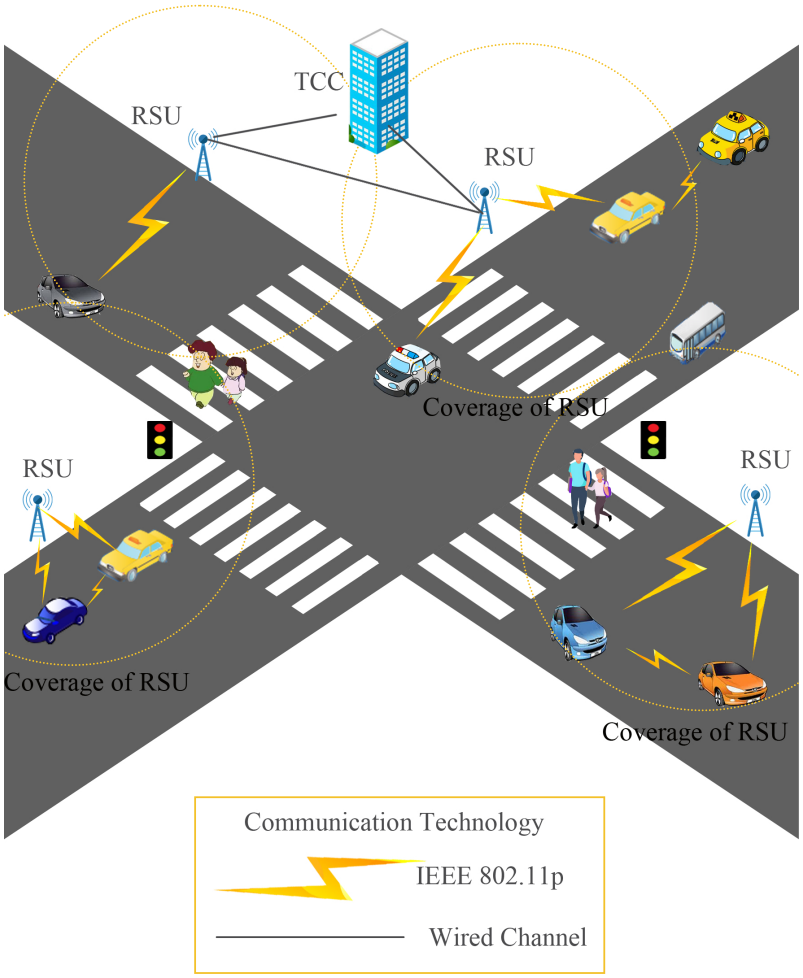

Figure 1. System architecture.

Figure 1 illustrates the system architecture within the scope of a single TCC. Communication between vehicles, as well as between vehicles and RSUs, is achieved through open dedicated short-range wireless signals, while communication between RSUs and between TCC and RSUs is performed through wired communication.

4.2. Definition of URSCS

Our URSCS system is mainly composed of the following ten algorithms, and we will introduce their basic definitions one-by-one. For the convenience of reading, the relevant symbols used in our URSCS scheme and their corresponding meanings are summarized in Table 1.

$(params, msk) \leftarrow Setup(k)$: On input security parameter k, the system initialization algorithm, which is run by the TCC, returns the public parameter $params$ and the system master secret key msk.

$(rsk, isk, IPK) \leftarrow VelRegister(RID)$: On input the real identity RID of the registration vehicle, the vehicle registration algorithm, which is run by the TCC, computes the revocation secret key rsk, identification secret key isk, and identification public key IPK for the registration vehicle.

Table 1. Definition of notations.

Notation	Definition
msk,MPK	System master secret key, system master public key
b	Master revocation secret key
$(B, batValue)$	Master revocation public key
V_j	The jth vehicle
RSU_j	The jth RSU
rsk_j	Revocation secret key of V_j
SK	Session key
isk_j	Identification secret key of V_j or RSU_j
IPK_j	Identification public key of V_j or RSU_j
(sk_j, PK_j)	Key pair of the entity with V_j
(y_j, Y_j)	Partial key pair of the entity with V_j
RID_j	Real Identity of V_j
PID_j	Pseudonym of V_j
m	Message
mod	Modulo operation
$H(.)$	Cryptographic hash functions
t	Timestamp
REV	Identification public key set of receivers
$\oplus$	XOR operation
$\|\|$	Concatenation of string
ε	Ciphertext in multiple recipients scenario.
ξ	Ciphertext in single recipient scenario.

$(isk, IPK) \leftarrow RSURegister(RID)$: On input the real identity RID of the registration RSU, the RSU registration algorithm, which is run by the TCC, generates the identification secret key isk and identification public key IPK for the registration RSU.

$(\{sk_j, PK_j\}_{j=1}^{n}) \leftarrow GenMK(RID)$: On input the real identity RID of the vehicle, the vehicle's key generation algorithm, which is run by the vehicles, generates a set of corresponding key pairs $\{sk_j, PK_j\}_{j=1}^{n}$.

$(\{PID_j\}_{j=1}^{n}, \{ppk_j\}_{j=1}^{n} \; or \; \perp) \leftarrow GenPIDandPPK(RID, \{PK_j\}_{j=1}^{n}, msk)$: On input the system master secret key msk, the real identity RID and a set corresponding public keys $\{PK_j\}_{j=1}^{n}$, the vehicle's pseudonyms and partial keys generation algorithm, which is run by the TCC, generates a set of corresponding pseudonyms $\{PID_j\}_{j=1}^{n}$ and a set of corresponding partial private keys $\{ppk_j\}_{j=1}^{n}$.

$(b, B, batVal) \leftarrow GenRovSK(count, \{rsk_j\}_{j=1}^{count})$: On input the number of all legal vehicles in the current TCC domain $count$ and the corresponding revocation keys for each legal vehicle $\{rsk_j\}_{j=1}^{count}$, the revocation key generation algorithm, which is run by the TCC, computes the revocation master key b, the corresponding master revocation public key $(B, batVal)$.

$(\varepsilon) \leftarrow MulSignCrypt(m, PID_s, sk_s, PK_s, y_s, Y_s, rsk_s, REV = \{IPK_j\}_{j=1}^{u})$: On input the message m, the pseudonym of the sender PID_s, the sender's keys: $sk_s, PK_s, y_s, Y_s, rsk_s$, and a set of receivers consisting of vehicles and RSUs REV, the multi-receiver signcryption algorithm, which is run by the RSUs or vehicles, generates a ciphertext ε.

$(m \; or \; \perp) \leftarrow MulDeSignCrypt(\varepsilon)$: On input a ciphertext ε, the multi-receiver designcryption algorithm, which is run by the RSUs or vehicles, outputs a message $m \; or \; \perp$.

$(\xi) \leftarrow SigSignCrypt(m, PID_s, sk_s, PK_s, y_s, Y_s, rsk_s, IPK_j)$: On input the message m, the pseudonym of the sender PID_s, the sender's keys: sk_s, PK_s, y_s, Y_s, the revocation key of the sender rsk_s, and the identity public key of the sender IPK_j, the single-receiver signcryption algorithm, which is run by vehicles, generates a ciphertext ξ.

$(m \; or \; \perp) \leftarrow SigDeSignCrypt(\xi)$: On input the ciphertext ξ, the single-receiver designcryption algorithm, which is run by the RSUs or vehicles, outputs a message $m \; or \; \perp$.

4.3. Security and Privacy Goals

In this section, we elaborate on the critical security and privacy goals that must be achieved to ensure the safe operation of VANETs. These goals are paramount as they directly impact the trustworthiness of VANETs.

Authentication: Ensuring that all communication participants, including vehicles and RSU, are legitimate. Specifically, receivers (including both vehicles and RSUs) must be able to validate that the sending entities (vehicles) are genuine and not impersonating other legitimate nodes.

Data Confidentiality: In the VANET, sensitive data should be transmitted exclusively in encrypted (ciphertext) form over open channels to minimize the risk of information disclosure. Only authorized recipients with the corresponding decryption key can access and view the original data, thereby ensuring confidentiality and preventing eavesdroppers from deciphering the content of intercepted messages.

Data Integrity: To guarantee the correctness and dependability of information transmitted across V2I and V2V in the VANETs, receivers (including both vehicles and RSUs) must be able to validate that the data have not been maliciously tampered with or altered in any way during transmission. This ensures that the data received is identical to the data sent, thereby maintaining data integrity.

Anonymity: Vehicles employ pseudonyms for message transmission, leveraging a secure anonymity system that safeguards against linking or identifying the actual cars or drivers behind the data. This approach ensures that, apart from the TCC, no other party can infer the real identities of vehicles based solely on their pseudonyms.

Unlinkability: In VANETs, the unlinkability ensures that there is no discernible linkage between multiple signcrypted messages transmitted by the same vehicle. This mechanism effectively obscures the connection between messages, preventing any entity from distinguishing or inferring whether two captured messages originate from the same sender. As a result, privacy is significantly enhanced, as the identity and actions of individual vehicles remain unlinkable within the network.

Traceability: In the case of a dispute arising or the detection of malicious activity, exclusive access to trace the actual identity of vehicles engaged in such behavior, based on their pseudonyms, is granted solely to the TCC.

Revocability: Upon detection of malicious behavior, TCC has the authority to revoke the access of offending vehicles from the system, effectively disabling their ability to transmit any further valid information.

Resistance to Various Attacks: Our URSCS scheme must exhibit resilience against a broad spectrum of known attack vectors, encompassing impersonation attacks, replay attacks, ephemeral secret leakage attacks, man-in-the-middle attacks, and other potential threats that pose risks to the system's integrity and confidentiality.

4.4. Security Model

According to the authors of [14], in certificateless signcryption schemes, there are two types of attackers. The *Type I* attacker can replace a user's public key but cannot obtain the system's master secret key. The *Type II* attacker possesses the system's master secret key but cannot replace a user's public key. In both cases, the signcryption scheme should achieve indistinguishability under adaptive chosen ciphertext attacks (IND-CCA2) and existential unforgeability against chosen message attacks (EU-CMA). For readability, we denote $\mathcal{A}_1^i$ as *Type I* attacker, where $\mathcal{A}_1^1$ attacks confidentiality, and $\mathcal{A}_1^2$ attacks unforgeability; we denote $\mathcal{A}_2^i$ as *Type II* attacker, where $\mathcal{A}_2^1$ attacks the confidentiality, and $\mathcal{A}_2^2$ attacks unforgeability.

4.4.1. Definition 1—Confidentiality (IND-CCA2-Secure)

If any PPT attackers $\mathcal{A}_1^1$ and $\mathcal{A}_2^1$ cannot win with a non-negligible advantage in the subsequent two games, Game *I* and Game *II*, then a URSCS scheme is IND-CCA2-Secure.

Game *I*: This game is played between an adversary $\mathcal{A}_1^1$ and a challenger $\mathcal{C}$. The specific process is as follows.

Initialization: Challenger $\mathcal{C}$ executes the initialization function $Setup(k)$, obtains the system parameters *params* and the system master secret key *msk*, $\mathcal{C}$ sends the *params* to adversary $\mathcal{A}_1^1$, and secretly retains the *msk*.

Phase 1: $\mathcal{C}$ selects a vehicle with the real identity RID^* as the sender and chooses a set of receivers with identification public key $REV^* = \{IPK_j^*\}_{j=1}^u$.

$\mathcal{C}$ sends the selected RID^* and $REV^* = \{IPK_j^*\}_{j=1}^u$ to opponent $\mathcal{A}_1^1$, then $\mathcal{A}_1^1$ submits some hash queries to challenger $\mathcal{C}$.

$\mathcal{A}_1^1$ submits some oracle queries, and $\mathcal{C}$ responds with relevant inquiry results, but these results cannot contain information related to the sender's private key and *msk*.

Challenge: $\mathcal{A}_1^1$ selects two messages m_1, m_2 of equal length and submits them to $\mathcal{C}$. Then, $\mathcal{C}$ randomly chooses a number $\beta \in \{0,1\}$ and performs the $MulSignCrypt(m_\beta, PID_s^*, sk_s^*, PK_s^*, y_s^*, Y_s^*, rsk_s^*, REV^* = \{IPK_j^*\}_{j=1}^u)$ function. Finally, $\mathcal{C}$ sends the signcryption ciphertext ε^* to $\mathcal{A}_1^1$.

Phase 2: After obtaining the signcryption ciphertext ε^*, $\mathcal{A}_1^1$ submits another set of oracle queries, excluding $MulDeSignCrypt(\varepsilon^*)$.

Guess: $\mathcal{A}_1^1$ outputs a guessed value β^*. If $\beta^* = \beta$, then $\mathcal{A}_1^1$ wins the game.

Probability Advantage: The probability advantage of $\mathcal{A}_1^1$ can be defined as: $Adv_{\mathcal{A}_1^1}^{IND-CCA2}(k) = |Pr[\beta^* = \beta] - 1/2|$.

Game II: This game is played between the adversary $\mathcal{A}_2^1$ and $\mathcal{C}$. The specific process is as follows.

Initialization: $\mathcal{C}$ executes the initialization function $Setup(k)$, obtains the system parameters *params* and the system master private key *msk*, $\mathcal{C}$ sends the them to adversary $\mathcal{A}_2^1$.

The intermediate process is similar to Game I, but $\mathcal{A}_2^1$ is not allowed to submit an oracle query to the $\mathcal{C}$ regarding the replacement of public keys.

Guess: $\mathcal{A}_2^1$ outputs a guessed value β^*. If $\beta^* = \beta$, then $\mathcal{A}_2^1$ wins the game.

Probability Advantage: The probability advantage of $\mathcal{A}_2^1$ can be defined as: $Adv_{\mathcal{A}_2^1}^{IND-CCA2}(k) = |Pr[\beta^* = \beta] - 1/2|$.

4.4.2. Definition 2—Unforgeability (EUF-CMA-Secure)

If any PPT attackers $\mathcal{A}_1^2$ and $\mathcal{A}_2^2$ cannot win with a non-negligible advantage in the subsequent two games, Game III and Game IV, then a URSCS scheme is EUF-CMA-Secure.

Game III: This game is played between the adversary $\mathcal{A}_1^2$ and $\mathcal{C}$. The specific process is as follows.

Initialization: Initialization is similar to Game I.

Inquiry: $\mathcal{A}_1^2$ submits oracle queries, and $\mathcal{C}$ returns relevant results.

Forgery: $\mathcal{A}_1^2$ outputs a forged ciphertext ε^* with RID^* and $REV^* = \{IPK_j^*\}_{j=1}^u$. If the forged ciphertext ε^* can be decrypted correctly by any recipient, $\mathcal{A}_1^2$ wins the game.

Probability Advantage: The probability advantage of $\mathcal{A}_1^2$ can be defined as: $Adv_{\mathcal{A}_1^2}^{EUF-CMA}(k) = Pr[MulDeSignCrypt(\varepsilon^*) \neq \perp]$.

Game IV: This game is played between the adversary $\mathcal{A}_2^2$ and $\mathcal{C}$. The specific process is as follows.

Initialization: Initialization is similar to Game II.

Inquiry: $\mathcal{A}_2^2$ submits oracle queries, and $\mathcal{C}$ returns relevant results.

Forgery: $\mathcal{A}_2^2$ outputs a forged ciphertext ε^* with RID^* and $REV^* = \{IPK_j^*\}_{j=1}^u$. If the forged ciphertext ε^* can be decrypted correctly by any recipient, $\mathcal{A}_2^2$ wins the game.

Probability Advantage: The probability advantage of $\mathcal{A}_2^2$ can be defined as: $Adv_{\mathcal{A}_2^2}^{EUF-CMA}(k) = Pr[MulDeSignCrypt(\varepsilon^*) \neq \perp]$.

5. The Proposed URSCS Scheme

5.1. Basic Algorithms

This section introduces the implementation details of basic algorithms one-by-one, which will be used to construct our URSCS scheme in Section 5.2.

(1) $(params, msk) \leftarrow Setup(k)$: Given a security parameter k, the TCC generates an elliptic curve G with a generator P and order q. It chooses a random element $msk \in Z_q^*$ as the system's master private key and generates the corresponding master public key $MPK = mskP$. Then, it generates six cryptographic hash functions: $H_1 \colon G \rightarrow \{0,1\}^{|ID|}$, $H_2 \colon G \times \{0,1\}^{|ID|} \times G \times G \rightarrow Z_q^*$, $H_3 \colon G \times G \times G \times \{0,1\}^{|ID|} \times G \rightarrow Z_q^*$, $H_4 \colon Z_q^* \times G \times \{0,1\}^{|ID|} \rightarrow \{0,1\}^{2|G|+|m|}$, $H_5 \colon G \times \{0,1\}^{|ID|} \times \{0,1\}^* \times \{0,1\}^m \times G \times G \times G \times Z_q^* \rightarrow Z_q^*$, and $H_6 \colon G \times G \times \{0,1\}^{|ID|} \rightarrow \{0,1\}^{2|G|+|m|}$, where $|ID|$ represents the length of the real ID and is determined by security parameter, $|G|$ is the length of the string representation of a point on the elliptic curve, and $|m|$ represents the length of a single message and is determined by security parameter. Then, TCC obtains the system's master secret key msk and the common parameters $params = \{G, P, MPK, H_1, H_2, H_3, H_4, H_5, H_6\}$.

(2) $(isk, IPK) \leftarrow RSURegister(RID)$: TCC chooses a random element $isk \in Z_q^*$ as the identification secret key of RSU and generates the corresponding identification public key $IPK = iskP$.

(3) $(rsk, isk, IPK) \leftarrow VelRegister(RID)$: When a vehicle with real identity RID requests registration, TCC first generates a revocation key rsk for the vehicle as follows: TCC maintains a pool of prime numbers $PoolPN$, each of which is unique, and when a vehicle needs to be assigned a revocation key, one is taken from the $PoolPN$ and assigned to it. Then, the real identity RID of the vehicle and its corresponding revocation key rsk are saved to the association table of TCC in the form of $(RID, rsk, true)$, where the value $true$ indicates that the current vehicle is legal. This mechanism allows TCC to quickly retrieve the revocation key of any vehicle when necessary, while maintaining a record of the vehicle's legitimacy status. When a vehicle is judged as malicious, its corresponding entity will be changed to $(RID, rsk, false)$, and the associated revocation key rsk is sent to the revocation pool $PoolREV$ with the form (rsk, t_{rev}), where t_{rev} indicates the timestamp of revocation. When a vehicle needs to be allocated a revocation key, but the prime numbers in $PoolPN$ have been exhausted, it is necessary to obtain a prime number from $PoolREV$, and the prime number needs to be checked to see if its revocation time exceeds threshold T_{rev}. If the revocation time has exceeded T_{rev}, it can be reused; otherwise, the process will continue to wait. Since the survival time of each pseudonym, T_{pid}, is less than T_{rev}, if a malicious vehicle's revocation key has been revoked for longer than T_{rev}, it means that the pseudonyms generated by the malicious vehicle before it was revoked are no longer available. This mechanism ensures that malicious vehicles cannot continue sending messages after their revocation keys have been reassigned to new vehicles. For more details, one can refer to Section 6.1.7. Subsequently, TCC selects a random number $isk \in Z_q^*$ as the identification secret key and generates the corresponding identification public key $IPK = iskP$.

(4) $(\{sk_j, PK_j\}_{j=1}^n) \leftarrow GenMK(RID)$: The vehicle with real identity RID selects n random numbers $\{sk_j\}_{j=1}^n \in Z_q^*$ as private keys, generates the corresponding n public keys $\{PK_j = sk_jP\}_{j=1}^n$, keeps $\{sk_j\}_{j=1}^n$ in secret, and publishes $\{PK_j\}_{j=1}^n$.

(5) $(\{PID_j\}_{j=1}^n, \{ppk_j\}_{j=1}^n \; or \; \perp) \leftarrow GenPIDandPPK(RID, \{PK_j\}_{j=1}^n, msk)$: The vehicle transmits its own real identity RID and a set of public keys $\{PK_j\}_{j=1}^n$ to TCC, and TCC can verify whether the vehicle is legal by checking the association table $(RID, rsk, true/false)$. If the value in the association table corresponding to the RID and rsk is $true$, then TCC performs the following procedure, otherwise TCC returns $\perp$.

- TCC generates a set of pseudonyms containing n elements $\{PID_j\}_{j=1}^n$ associated with RID, each of which is structured as $PID_j = (PID_j^1, PID_j^2, t_j^{pid})$, where t_j^{pid} is the timestamp used to control the lifetime of PID, $PID_j^1 = x_jP$, $PID_j^2 = RID \oplus H_1(mskPID_j^1)$, and $x_j \in Z_q^*$.

- TCC randomly chooses n random elements $z_j \in Z_q^*$, obtains n points $Y_j = z_j P$ on the corresponding elliptic curve, and then computes $h_j = H_2(PID_j, Y_j, PK_j)$ and $y_j = z_j + h_j msk$, for $j = 1, \cdots, n$, hence, TCC obtains a set of partial public-private key pairs $\{ppk_j = (y_j, Y_j)\}_{j=1}^{n}$.

(6) $(b, B, batVal) \leftarrow GenRovSK(count, \{rsk_j\}_{j=1}^{count})$: The $count$ here represents the number of existing legal vehicles in the TCC domain, and $\{rsk_j\}_{j=1}^{count}$ are the corresponding revocation secret keys for each legal vehicle. When there are vehicles joining or exiting, the value $count$ would be automatically changed accordingly. TCC first selects a random prime number $b \in PoolPN$ as the master revocation secret key and then removes b from the $PoolPN$. When the system needs to be allocated a master revocation secret key, but the prime numbers in $PoolPN$ have been exhausted, the handling mechanism employed is identical to the *VelRegister* method. TCC generates $B = bP$ and utilizes CRT to construct other parts of the master revocation public key, calculating $\alpha = \prod_{j=1}^{count} rsk_j$, $\alpha_j = \alpha / rsk_j$. TCC computes the multiplicative inverse θ_j of α_j modulo rsk_j, satisfying $\theta_j \alpha_j \equiv 1 \ mod \ (rsk_j)$. Since rsk_j are different prime numbers, rsk_j and α_j are coprime, which implies that the multiplicative inverse θ_j of α_j modulo rsk_j exists. Next, TCC calculates $Sum = \sum_{j=1}^{count} \theta_j \alpha_j$ and $batVal = bSum$. Finally, TCC secretly keeps the master revocation secret key b and makes $(B, batVal)$ public.

(7) $(\varepsilon) \leftarrow MulSignCrypt(m, PID_s, sk_s, PK_s, y_s, Y_s, rsk_s, REV = \{IPK_j\}_{j=1}^{u})$: Assume that vehicle V_s needs to send message m to a group of entities REV. Vehicle V_s selects a set of unused $(PID_s, sk_s, PK_s, y_s, Y_s)$, and performs the following calculations, removing these parameters from the vehicle parameter pool. The vehicle calculates the revocation master key $b = batVal \ mod \ rsk_s$, which is equivalent to the master revocation secret key b that is selected by TCC. The equivalence can be validated as follows.

$$\begin{aligned} b &= batVal \ mod \ rsk_s \\ &= (bSum) \ mod \ rsk_s \\ &= (b \sum_{j=1}^{count} \theta_j \alpha_j) \ mod \ rsk_s \\ &= (b(\theta_1 \prod_{j=2}^{count} rsk_j + \theta_2 \prod_{j=1, j\neq 2}^{count} rsk_j + \cdots + \theta_s \\ &\prod_{j=1, j\neq s}^{count} rsk_j + \cdots + \theta_{count} \prod_{j=1}^{count-1} rsk_j)) \ mod \ rsk_s \\ &= (b(\theta_s \prod_{j=1, j\neq s}^{count} rsk_j)) \ mod \ rsk_s \\ &= b. \end{aligned} \tag{2}$$

In Equation (2), the values of the intermediate variables are set at the TCC side as follows: $\alpha = \prod_{j=1}^{count} rsk_j$, $\alpha_j = \alpha / rsk_j$, $\theta_j \alpha_j \equiv 1 \ mod \ (rsk_j)$, $sum = \sum_{j=1}^{count} \theta_j \alpha_j$, $batVal = bsum$. Obviously, the master revocation secret key b calculated by the sending vehicle is the same as b set by the TCC.

Then, vehicle V_s selects a random number $r \in Z_q^*$, and calculates $R = rP$. For each identity public key $IPK_j \in REV$, vehicle V_s calculates $\omega_j = (r + b)IPK_j$ and $k_j = H_3(\omega_j, IPK_j, PID_s, R)$. Then, it randomly selects a session key $SK \in Z_q^*$, constructs a polynomial $f(x) = \prod_{j=1}^{u}(x - k_j) + SK = a_0 + a_1 x^1 + a_2 x^2 + \cdots + a_{u-1} x^{u-1} + x^u$, and calculates the ciphertext $C = H_4(SK, PID_s) \oplus (PK_s || Y_s || m)$. The vehicle then selects a timestamp t and calculates $d_1 = H_5(PID_s, t, m, PK_s, Y_s, R, 1)$, $d_2 = H_5(PID_s, t, m, PK_s, Y_s, R, 2)$, $d_3 = H_5(PID_s, t, m, PK_s, Y_s, R, 3)$, $\rho = r + d_1 y_s + d_2 sk_s + d_3 b$. The vehicle can obtain the signcrypted information $\varepsilon = (PID_s, C, \rho, t, R, a_0, a_1, \ldots, a_{u-1})$, which is broadcast to all entities, including many vehicles or RSUs.

(8) $(m \text{ or } \perp) \leftarrow MulDeSignCrypt(\varepsilon)$: Assuming that vehicle V_j receives the signcrypted message ε, it first checks the validity of the timestamp t. If the timestamp has expired, it returns $\perp$, rejecting the signcrypt information. If the timestamp is valid, the vehicle obtains the timestamp t_s^{pid} from PID_s, and then verifies whether the existence time of the PID_s exceeds threshold T_{pid}. If the existence time exceeds T_{pid}, it returns $\perp$; otherwise, it continues to perform the following operations. Note that T_{pid} should be less than T_{rev} to prevent malicious vehicles from continuing to send messages after their revocation keys are reassigned to the new vehicles.

Vehicle V_j uses the master revocation public key B, its own identity private key isk_j, the identity public key IPK_j, R and PID_s from the signcrypted message to calculate $\omega_j = isk_j(R + B)$, $k_j = H_3(\omega_j, IPK_j, PID_s, R)$, and constructs a polynomial $f(x) = a_0 + a_1x^1 + a_2x^2 + \cdots + a_{u-1}x^{u-1} + x^u$ according to the received elements $a_0, a_1, \ldots, a_{u-1}$. Then vehicle V_j calculates the session key $SK = f(k_j)$, $(PK_s, Y_s, m) = H_4(SK, PID_s) \oplus C$, and computes $d_1 = H_5(PID_s, t, m, PK_s, Y_s, R, 1)$, $d_2 = H_5(PID_s, t, m, PK_s, Y_s, R, 2)$, $d_3 = H_5(PID_s, t, m, PK_s, Y_s, R, 3)$, $h_s = H_2(PID_s, Y_s, PK_s)$. Vehicle V_j verifies if Equation (3) holds.

$$\rho P = R + d_1Y_s + d_1h_sMPK + d_2PK_s + d_3B \tag{3}$$

If the equality in (3) holds, it accepts the message m; otherwise, it rejects the message. If the target receiving vehicle is the same as the current receiving vehicle V_j, the calculated values will be the same as those of the sending vehicle. The correctness of Equation (3) can be validated as follows.

$$\begin{aligned} L.H.S. &= (r + d_1y_s + d_2sk_s + d_3b)P \\ &= rP + d_1y_sP + d_2sk_sP + d_3bP \\ &= R + d_1y_sP + d_2PK_s + d_3B \\ &= R + d_1z_sP + d_1h_smskP + d_2PK_s + d_3B \\ &= R + d_1Y_s + d_1h_sMPK + d_2PK_s + d_3B \\ &= R.H.S. \end{aligned} \tag{4}$$

In Equation (4), the intermediate variables were set at the sending side as follows: $\rho = r + d_1y_s + d_2sk_s + d_3b$, $B = bP$, $R = rP$, $y_s = z_s + h_smsk$, $h_s = H_2(PID_s, Y_s, PK_s)$, $Y_s = z_sP$, and the values d_1, d_2, and d_3 on the sending side are: $d_1 = H_5(PID_s, t, m, PK_s, Y_s, R, 1)$, $d_2 = H_5(PID_s, t, m, PK_s, Y_s, R, 2)$, $d_3 = H_5(PID_s, t, m, PK_s, Y_s, R, 3)$.

(9) $(\xi) \leftarrow SigSignCrypt(m, PID_s, sk_s, PK_s, y_s, Y_s, rsk_s, IPK_r)$: Assume that vehicle V_s needs to send message m to a specific vehicle or RSU. The vehicle calculates the revocation master key $b = batVal \bmod rsk_s$, which would be equivalent to the master revocation secret key b that is selected by TCC. Then, it selects a random number $r \in Z_q^*$ and calculates $R = rP$. Next, vehicle V_s selects a set of unused parameters $(PID_s, sk_s, PK_s, y_s, Y_s)$, performs the following steps, and removes these parameters from the vehicle parameter pool. Using the identity public key IPK_j of the receiving vehicle or RSU, it calculates $\omega_j = (r + b)IPK_j$, and computes the ciphertext $C = H_6(\omega_j, PID_s) \oplus (PK_s||Y_s||m)$. The vehicle then selects a timestamp t and calculates $d_1 = H_5(PID_s, t, m, PK_s, Y_s, R, 1)$, $d_2 = H_5(PID_s, t, m, PK_s, Y_s, R, 2)$, $d_3 = H_5(PID_s, t, m, PK_s, Y_s, R, 3)$, $\rho = r + d_1y_s + d_2sk_s + d_3b$. The vehicle can obtain the signcryption information $\xi = (PID_s, C, \rho, t, R)$, which is broadcast to all entities.

(10) $(m \text{ or } \perp) \leftarrow SigDeSignCrypt(\xi)$: Assuming that vehicle V_j receives the signcrypted message ξ, it first checks the validity of the timestamp t. If the timestamp has expired, it returns $\perp$, rejecting the signcrypt information. If the timestamp is valid, the vehicle obtains the timestamp t_s^{pid} from PID_s, and then verifies whether the existence time of the PID_s exceeds threshold T_{pid}. If the existence time exceeds T_{pid}, it returns $\perp$; otherwise, it continues to perform the following steps. The vehicle uses the master revocation public key B, its own identification private key isk_j, the identification public key IPK_j, the random point R and PID_s from the signcrypted message to calculate $\omega_j = isk_j(R +$

$B)$, $(PK_s||Y_s||m) = H_6(\omega_j, PID_s) \oplus C$, and computes $d_1 = H_5(PID_s, t, m, PK_s, Y_s, R, 1)$, $d_2 = H_5(PID_s, t, m, PK_s, Y_s, R, 2)$, $d_3 = H_5(PID_s, t, m, PK_s, Y_s, R, 3)$, $h_s = H_2(PID_s, Y_s, PK_s)$. Vehicle V_j verifies if Equation (5) holds.

$$\rho P = R + d_1 Y_s + d_1 h_s MPK + d_2 PK_s + d_3 B \tag{5}$$

If true, it accepts the message m; otherwise, it rejects the message. The validity of Equation (5) can be proved as in Equation (4).

5.2. URSCS Construction

This section utilizes the algorithms detailed in the previous section to form our URSCS scheme. It consists of five stages, namely, initialization, registration, signcryption, designcryption, as well as trace and revocation.

5.2.1. Initialization

First, TCC executes the system initialization algorithm $Setup(k)$, obtaining the system master key msk and system public parameters $params = \{G, P, MPK, H_1, H_2, H_3, H_4, H_5, H_6\}$. Next, it publicly discloses $params$ and secretly holds the system master key msk.

5.2.2. Registration

The vehicle sends its real identity RID to TCC to request registration. TCC executes the $VelRegister(RID)$ algorithm, generates the identification secret key isk, the identification public key IPK, and the revocation key rsk. TCC sends isk, IPK, and rsk to the registering vehicle through a secure channel. The vehicle secretly stores isk and rsk within the tamper-proof device of the OBU and publishes the identification public key IPK. RID cannot be disclosed, so the IPK is used instead of the RID as the long-term identity and made public to all entities.

The RSU sends its RID to TCC to request registration. TCC executes the $RSURegister(RID)$ algorithm, generates the identification secret key isk and the identification public key IPK. TCC then sends isk and IPK to the registering RSU through a secure channel. The RSU secretly keeps the isk and publishes the IPK. In the multi-receiver signcryption scenario, the receivers can be either vehicles or RSUs. To unify the identifiers of the receivers, the identification public keys of the RSUs are used to replace their real IDs.

The registered vehicle executes the vehicle key generation algorithm $GenMK(RID)$, generating a set of public-private key pairs: $(\{sk_j\}_{j=1}^n, \{PK_j\}_{j=1}^n)$. The vehicle sends RID and $\{PK_j\}_{j=1}^n$ to the TCC to request generation of pseudonyms and partial keys. Upon receiving this request, TCC executes the $GenPIDandPPK(RID, \{PK_j\}_{j=1}^n, msk)$ algorithm, verifies the legitimacy of the requesting vehicle, then generates a set of pseudonyms $\{PID_j\}_{j=1}^n$ and a set of partial keys $\{ppk_j\}_{j=1}^n$, and sends these pseudonyms and partial keys to the requesting vehicle through a secure channel. And then, to facilitate information retrieval, the vehicle correctly matches each pseudonym with its corresponding public-private key pair and partial key pair, and stores them in a secure pool as $(PID_j, sk_j, PK_j, ppk_j)$. To ensure unlinkability, the vehicle consumes one of these tuples $(PID_j, sk_j, PK_j, ppk_j)$ each time it sends a message.

TCC executes the revocation key generation algorithm $GenRovSK(count, \{rsk_j\}_{j=1}^{count})$, thereby obtaining the current system's master revocation secret key b, master revocation public key B, and the auxiliary value $batVal$. Finally, TCC secretly keeps the master revocation secret key b and publishes $(B, batVal)$.

5.2.3. Signcryption

Vehicles can signcrypt messages for multiple recipients or a single recipient.

In the multiple recipients scenario, each time a vehicle sends a message, it first selects a pseudonym and a key tuple $(PID_s, sk_s, PK_s, y_s, Y_s)$ from the secure pool, and removes this

pseudonym and its related key pair from the pool. When the pseudonyms and key tuples in the pool are about to be depleted, the vehicle re-executes the $GenMK(RID)$ algorithm and requests TCC to execute the $GenPIDandPPK(RID, \{PK_j\}_{j=1}^{n}, msk)$ algorithm to replenish the pool. The vehicle selects specific recipients $REV = \{IPK_j\}_{j=1}^{u}$, and executes the multi-recipient signcryption algorithm $MulSignCrypt(m, PID_s, sk_s, PK_s, y_s, Y_s, rsk_s, REV)$ to obtain the signcrypted information ε, and broadcasts it to multiple recipients.

In the single-recipient scenario, each time a vehicle sends a message, it first selects a pseudonym and a key tuple $(PID_s, sk_s, PK_s, y_s, Y_s)$ from the secure pool. The vehicle selects a specific recipient and executes the single-recipient signcryption algorithm $SigSignCrypt(m, PID_s, sk_s, PK_s, y_s, Y_s, rsk_s, IPK_j)$, obtains and broadcasts the signcrypted information ξ. The management of pseudonyms and keys in the secure pool is the same as in the multi-recipient signcryption scenario.

5.2.4. Designcryption

The designcryption stage also includes multi-recipient designcryption and single-recipient designcryption.

In multi-recipient designcryption, only the recipient specified by the sender can obtain the correct session key SK and successfully designcryption the message if $\omega_j = isk_j(R + B)$ calculated by the recipient matches $\omega_j = (r + b)IPK_j$ calculated by the sender.

Single-recipient designcryption shares the same principles as in multi-recipient designcryption and is omitted here.

5.2.5. Trace and Revocation

If an RSU or vehicle detects suspicious behavior from a malicious vehicle, it can submit the corresponding pseudonym $PID_f = (PID_f^1, PID_f^2, t_f^{pid})$ to TCC. Then, TCC calculates $RID_f = PID_f^2 \oplus H_1(mskPID_f^1)$ using the system master secret key msk and the pseudonym PID_f to obtain RID_f of the malicious vehicle.

Then, TCC receives the revocation secret key rsk_f from the association pair $(RID_f, rsk_f, true)$ of the association table, removes the rsk_f of the malicious vehicle from the revocation key set $\{rsk_j\}_{j=1}^{count}$, and adds it to $PoolREV$ as (rsk_f, t_{rev}), then changes the association pair $(RID_f, rsk_f, true)$ to $(RID_f, rsk_f, false)$. TCC then randomly selects a new master revocation secret key b^{new} and calculates $B^{new} = b^{new}P$. By employing CRT, TCC re-executes the procedure of generating the remaining part of the master revocation public key, the specific implementation procedure can be found in the $GenRovSK$ algorithm described in Section 5.1. Note that the revocation secret key of the malicious vehicle has been deleted, so $\alpha = \prod_{j=1}^{count-1} rsk_j$, where $count$ represents the number of legitimate vehicle. Finally, TCC secretly keeps the new master revocation secret key b^{new} and publishes the new master revocation public key $(B^{new}, batVal^{new})$. The vehicle whose revocation secret key rsk_f has been deleted cannot obtain the latest master revocation secret key b^{new} using $batVal^{new}$ mod rsk_f, but other normal vehicles can.

6. Security Analysis

In this section, we provide the informal security analysis, and formal security proof of the proposed URSCS scheme.

6.1. Informal Security Analysis

We show that the proposed scheme URSCS meets the security and privacy goals proposed in Section 4.3.

6.1.1. Authentication

In the proposed URSCS scheme, only the legitimate vehicle can obtain revocation secret key rsk through registration; furthermore, only the legitimate vehicle can obtain the master revocation secret key b using rsk and send messages. From another perspective,

only legitimate vehicles can obtain their own secret keys and compute ρ using their own secret keys, and receivers can verify the validity of ρ using the sender's public key, the system's master public key, and the system's master revocation public key. The formal security analysis, Theorems 3 and 4 , also demonstrate the unforgeability of signatures under the CDHP. Therefore, our URSCS scheme achieves vehicle authentication, ensuring that senders are legitimate vehicles in the system.

6.1.2. Data Confidentiality

The confidentiality of message m is protected by the session key SK. The sender encrypts m as $C = H_4(SK, PID_s) \oplus (PK_s||Y_s||m)$, and only the legitimate recipient can compute the session key SK, and then obtain the decrypted message $(PK_s||Y_s||m) = H_4(SK, PID_s) \oplus C$. Additionally, formal security proofs of Theorems 1 and 2 would demonstrate that if the CDHP is hard to solve, then malicious users cannot obtain the real data, ensuring the confidentiality of the data.

6.1.3. Data Integrity

As shown in Section 5.1, the recipient needs to calculate the hash values d_1, d_2, and d_3 before receiving the message in designcryption process. Since d_1, d_2, and d_3 are calculated in the same manner at both the sender's and receiver's ends, for example, the calculation method for d_1 at both ends is $d_1 = H_5(PID_s, t, m, PK_s, Y_s, R, 1)$. The message m is one of the inputs to the hash function H_5. Suppose the message m is tampered with during transmission, the hash values d_1, d_2, d_3 calculated at the sender's end will be different from those calculated by the recipient, and the verification process will fail. Therefore, our scheme ensures data integrity.

6.1.4. Anonymity

In our URSCS scheme, a pseudonym mechanism is employed to achieve sender anonymity. To extract RID, an attacker needs to calculate $RID = PID^2 \oplus H_1(mskPID^1)$; therefore, the attacker must obtain msk or solve the ECDLP problem. Since only TCC knows the system's master secret key msk, and the ECDLP is hard to solve, the adversary cannot infer the real RID from PID. Thus, our scheme achieves sender anonymity.

On the other hand, the identification public key IPK is used to replace real RID for message reception. Other than the sender, no other entity can infer the message recipient from intercepted ciphertext messages. Therefore, our URSCS achieves receiver anonymity.

6.1.5. Unlinkability

Our URSCS scheme achieves anonymity by hiding the RID under PID. In generating pseudonyms, each pseudonym is calculated using a random number x. When a vehicle performs the signcryption function, the random numbers SK, r, t are used; thus, no adversary can link any vehicle through pseudonyms or ciphertexts from the same sender. Furthermore, even if the malicious vehicles obtain the public keys from the ciphertexts, they cannot link to the vehicle through the public keys because the vehicle needs to fetch a new pseudonym and the corresponding public-private key from the security pool before each signcryption operation. Therefore, our URSCS achieves unlinkability.

6.1.6. Traceability

In the event of a dispute or the discovery of malicious behavior, TCC can obtain the true identity of the vehicle through the equation $RID_f = PID_f^2 \oplus H_1(mskPID_f^1)$, where RID_f is the real identity of the malicious vehicle and $PID_f = (PID_f^1, PID_f^2, t_f^{pid})$ is the pseudonym of the malicious vehicle. Thus, our URSCS system can achieve tracking functionality while maintaining the anonymity of vehicles.

6.1.7. Revocability

After tracking the malicious vehicle V_f, TCC removes the revocation key rsk_f of V_f from the revocation secret key set $\{rsk_j\}_{j=1}^{count}$ and adds it to $PoolREV$. Then, TCC randomly selects a new master revocation secret key b^{new} and calculates $B^{new} = b^{new}P$. Next, TCC uses the CRT to construct the other part of the master revocation public key, and calculates $\alpha = \prod_{j=1}^{count-1} rsk_j$; thus, the revocation secret key of V_f has been removed. Then, TCC calculates $\alpha_j = \alpha / rsk_j$, finds the multiplicative inverse θ_j of $\alpha_j \ mod \ rsk_j$, and computes $Sum = \sum_{j=1}^{count-1} \theta_j \alpha_j$, $batVal^{new} = b^{new} Sum$. Finally, the TCC retains the master revocation secret key b^{new} secretly and publishes the master revocation public key $(B^{new}, batVal^{new})$. V_f can no longer obtain the latest master revocation secret key b^{new} through $batVal^{new} \ mod \ rsk_f$; thus, V_f will be unable to send valid messages again.

In our URSCS scheme, if rsk_f is reallocated to a new vehicle, it implies that the existing time of rsk_f in the revocation pool $PoolREV$ has exceeded threshold T_{rev}, as the threshold T_{pid} is less than the threshold T_{rev}; therefore, the pseudonyms of V_f cannot be used, and this method is used to prevent malicious vehicles from continuing to send messages after their revocation keys are reassigned to the new vehicles.

6.1.8. Resistance to Impersonation Attacks

In our URSCS scheme, the malicious vehicle V_f with real identity RID_f can intercept the master revocation public key B, and the ciphertext $\varepsilon = (PID_s, C, \rho, t, R)$, but it cannot obtain the private key (y_s, sk_s) of vehicle V_s with the pseudonym PID_s. If the malicious vehicle wants to impersonate vehicle V_s, it may attempt to use its key pair (sk_f, PK_f, y_f, Y_f) and the pseudonym PID_s of a legitimate vehicle V_s to send information, since $y_f = z_f + H_2(PID_f, Y_f, PK_f)msk$ (which contains the pseudonym PID_f of the malicious vehicle). However, the malicious vehicle passed PID_s to the recipients, the receivers need to compute $h_s = H_2(PID_s, Y_f, PK_f)$, and this will not satisfy the verification Equation (3). Therefore, our URSCS scheme can resist impersonation attacks.

6.1.9. Resistance to Replay Attacks

In our scheme, each transmitted ciphertext $\varepsilon = (PID_s, C, \rho, t, R)$ contains a timestamp t. After receiving these messages, the receiver can determine if a replay attack has occurred by checking their timestamps. Therefore, our scheme can resist replay attacks.

6.1.10. Resistance to Ephemeral Secret Leakage Attacks

In our scheme, the session key SK is randomly chosen by the sender for each transmission. Due to the randomness of the session key and the lack of correlation between the session keys chosen for different transmissions, even if a specific session key is disclosed, it does not threaten the already created session keys or those to be created. Therefore, our scheme ensures the freshness of the session keys, can resist ephemeral secret leakage attacks, and also possesses forward and backward security.

6.1.11. Resistance to Man-in-the-Middle Attacks

Since our scheme achieves authentication and integrity guarantee, it can resist man-in-the-middle attacks.

In the same manner, we have analyzed the security and privacy goals achieved by schemes [8,13,17,18] . Table 2 elaborates on the comparison of these properties between our USS-MR scheme and similar schemes, namely [8,13,17,18]. The identifiers Resis-Imper, Resis-Replay, Resis-ESL, and Resis-Man represent resistance to impersonation attacks, resistance to replay attacks, resistance to ephemeral secret leakage attacks, and resistance to man-in-the-middle attacks, respectively. As shown in Table 2, only our URSCS scheme and scheme [18] can meet all security and privacy goals.

Table 2. Comparison of security properties.

Security Indicators	[8]	[13]	[17]	[18]	Ours
Authentication	✓	✓	✓	✓	✓
Confidentiality	✓	✓	✓	✓	✓
Integrity	✓	✓	✓	✓	✓
Vehicle Anonymity	✓	✗	✓	✓	✓
Receiver Anonymity	✓	✓	✓	✓	✓
Unlinkability	✓	✗	✓	✓	✓
Traceability	✓	✗	✗	✓	✓
Revocability	✗	✗	✗	✓	✓
Resis-Imper	✓	✓	✓	✓	✓
Resis-Replay	✓	✓	✗	✓	✓
Resis-ESL	✓	✓	✓	✓	✓
Resis-Man	✓	✓	✓	✓	✓

✓ indicates that the security function is enabled; ✗ that the security function is not enabled.

6.2. Formal Security Proof

6.2.1. Confidentiality

Our URSCS scheme achieves confidentiality as shown in Theorems 1 and 2.

Theorem 1. *Assume that $\mathcal{A}_1^1$ can win Game I with non-negligible advantage δ, then $\mathcal{C}$ can solve CDHP with non-negligible advantage δ'.*

Proof of Theorem 1. Assume that challenger $\mathcal{C}$ obtains an instance of the CDHP $(P, \kappa P, \tau P)$ and attempts to compute $\kappa\tau P$. The interaction between $\mathcal{C}$ and $\mathcal{A}_1^1$ is as follows.

Initialization: $\mathcal{C}$ executes the initialization algorithm $Setup(k)$ to generate system parameters $params = \{G, P, MPK, H_1, H_2, H_3, H_4, H_5\}$, where $MPK = \kappa P$, and sends the system parameters $params$ to $\mathcal{A}_1^1$.

Phase 1: $\mathcal{C}$ selects a sender with real identity RID^* and u receivers with real identities $REVIDs^* = \{RID_j^*\}_{j=1}^u$ for $\mathcal{A}_1^1$. $\mathcal{C}$ executes the algorithm $VelRegister(RID_j^*)$ or $RSURegister(RID_j^*)$ where $j \in \{1, 2, \ldots, u\}$ to obtain $\{isk_j^*, IPK_j^*\}_{j=1}^u$. $\mathcal{C}$ publishes the receivers' identification public keys $REV^* = \{IPK_j^*\}_{j=1}^u$ and keeps the private keys of the receivers privately.

$\mathcal{C}$ executes the algorithm $GenMK(RID^*)$ to obtain $\{sk_j^*, PK_j^*\}_{j=1}^n$ and executes the algorithm $GenPIDandPPK(RID^*, \{PK_j^*\}_{j=1}^n, msk)$ to obtain $\{PID_j^*, PPK_j^*\}_{j=1}^n$. $\mathcal{C}$ keeps five lists L_{H1}, L_{H2}, L_{H3}, L_{H4}, which are initially empty, and L_{H5}. Then, $\mathcal{A}_1^1$ adaptively conducts a polynomially bounded number of queries as follows.

H_1 Query : $\mathcal{C}$ receives PID_j^*. If there exists a corresponding tuple (PID_j^{1*}, λ_j) in L_{H1}, $\mathcal{C}$ sends λ_j to $\mathcal{A}_1^1$; otherwise, $\mathcal{C}$ selects $\lambda_j \in \{0,1\}^{|ID|}$ and sends λ_j to $\mathcal{A}_1^1$. Also, the tuple (PID_j^{1*}, λ_j) is inserted into the list L_{H1}.

H_2 Query : $\mathcal{C}$ receives $\{PID_j^*, Y_j^*, PK_j^*\}$. If there exists a corresponding tuple $(PID_j^*, Y_j^*, PK_j^*, h_j)$ in L_{H2}, $\mathcal{C}$ sends h_j to $\mathcal{A}_1^1$; otherwise, $\mathcal{C}$ selects $h_j \in Z_q^*$ and sends h_j to $\mathcal{A}_1^1$. Also, the tuple $(PID_j^*, Y_j^*, PK_j^*, h_j)$ is inserted into the list L_{H2}.

H_3 Query : $\mathcal{C}$ receives $\{\omega_r^*, IPK_r^*, PID_j^*, R\}$. If there exists a corresponding tuple $(\omega_r^*, IPK_r^*, PID_j^*, R, k_j)$ in L_{H3}, $\mathcal{C}$ sends k_j to $\mathcal{A}_1^1$; otherwise, $\mathcal{C}$ selects $k_j \in Z_q^*$ and sends k_j to $\mathcal{A}_1^1$. Also, the tuple $(\omega_r^*, IPK_r^*, PID_j^*, R, k_j)$ is inserted into the list L_{H3}.

H_4 Query : $\mathcal{C}$ receives $\{SK, PID_j^*\}$. If there exists a corresponding tuple (SK, PID_j^*, μ_j) in L_{H4}, $\mathcal{C}$ sends μ_j to $\mathcal{A}_1^1$; otherwise, $\mathcal{C}$ selects $\mu_j \in \{0,1\}^{2|G|+|M|}$ and sends μ_j to $\mathcal{A}_1^1$. Also, the tuple (SK, PID_j^*, μ_j) is inserted into the list L_{H4}.

H_5 Query : $\mathcal{C}$ receives $\{PID_j^*, t, m, PK_j^*, Y_j^*, R\}$. If there exists a corresponding tuple $(PID_j^*, t, m, PK_j, Y_j, R, d_1, d_2, d_3)$ in L_{H5}, $\mathcal{C}$ sends (d_1, d_2, d_3) to $\mathcal{A}_1^1$; otherwise, $\mathcal{C}$ selects $d_1 \in Z_q^*, d_2 \in Z_q^*, d_3 \in Z_q^*$ and sends (d_1, d_2, d_3) to $\mathcal{A}_1^1$. Also, the tuple $(PID_j^*, t, m, PK_j, Y_j, R, d_1, d_2, d_3)$ is inserted into the list L_{H5}.

$\mathcal{C}$ keeps three empty lists L_{rk}, L_{vk}, and L_{ppk}. Next, $\mathcal{C}$ responds to hash oracles queries from $\mathcal{A}_1^1$ as follows.

revokeKeyExtractQuery: Upon receiving RID, if $RID = RID^*$, $\mathcal{C}$ terminates the game; otherwise, $\mathcal{C}$ checks if there exists a corresponding tuple (RID, isk, IPK, rsk) in L_{rk}. If true, $\mathcal{C}$ sends rsk to $\mathcal{A}_1^1$; otherwise, $\mathcal{C}$ randomly selects a prime number rsk and sends it to $\mathcal{A}_1^1$. Also, $\mathcal{C}$ inserts the tuple (RID, isk, IPK, rsk) into L_{rk}.

secretValueExtractQuery: Upon receiving RID, if $RID = RID^*$, $\mathcal{C}$ terminates the game; otherwise, $\mathcal{C}$ checks if there exists a corresponding tuple (RID, sk_j, PK_j) in L_{vk}. If true, $\mathcal{C}$ sends sk_j to $\mathcal{A}_1^1$; otherwise, $\mathcal{C}$ randomly selects a $sk_j \in Z_q^*$, calculates $PK_j = sk_jP$, and sends sk_j to $\mathcal{A}_1^1$. Also, $\mathcal{C}$ inserts the tuple (RID, sk_j, PK_j) into L_{vk}.

publicValueExtractQuery: Upon receiving RID, if $RID = RID^*$, $\mathcal{C}$ terminates the game; otherwise, $\mathcal{C}$ checks if there exists a corresponding tuple (RID, sk_j, PK_j) in L_{vk}. If such a tuple exists, $\mathcal{C}$ sends PK_j to $\mathcal{A}_1^1$; otherwise, $\mathcal{C}$ randomly selects a $sk_j \in Z_q^*$, calculates $PK_j = sk_jP$, and sends PK_j to $\mathcal{A}_1^1$. Also, $\mathcal{C}$ inserts the tuple (RID, sk_j, PK_j) into L_{vk}.

partialPrivateKeyQuery: Upon receiving RID, if $RID = RID^*$, $\mathcal{C}$ terminates the game; otherwise, $\mathcal{C}$ checks if there exists a corresponding tuple $(RID, PK_j, PID_j, ppk_j)$ in L_{ppk}. If such a tuple exists, $\mathcal{C}$ sends ppk_j to $\mathcal{A}_1^1$; otherwise, $\mathcal{C}$ randomly selects $z_j \in Z_q^*$ and computes $Y_j = z_jP$, randomly selects $PID \in G$, then $\mathcal{C}$ executes H_2 *Query* to obtain h_j, and sends $ppk_j = (y_j = z_j + \kappa h_j, Y_j)$ to $\mathcal{A}_1^1$. Also, $\mathcal{C}$ inserts the tuple $(RID, PK_j, PID_j, ppk_j)$ into L_{ppk}.

replacePublicKeyQuery: $\mathcal{A}_1^1$ randomly selects (Y_j, PK_j) within an appropriate range to replace the complete public key (Y_j', PK_j') of identity RID. If $RID = RID^*$, then $\mathcal{C}$ stops the game; otherwise, $\mathcal{C}$ updates the list L_{vk} with (RID, sk_j, PK_j'), and updates the list L_{ppk} with $(RID, PK_j, PID_j, ppk_j')$.

mulSignCryptQuery: Upon receiving PID_j, $REV^* = \{IPK_j^*\}_{j=1}^u$ and m, $\mathcal{C}$ makes the following response:

(a) If $PID_j \neq PID_j^*$, challenger $\mathcal{C}$ makes *revokeKeyExtractQuery*, *secretValueExtractQuery*, *publicValueExtractQuery*, and *partialPrivateKeyQuery* to obtain (sk_j, PK_j, ppk_j, rsk). Next, $\mathcal{C}$ runs $MulSignCrypt(m, PID_s, sk_s, PK_s, y_s, Y_s, rsk_s, REV = \{IPK_j\}_{j=1}^u)$ to get $\varepsilon = (PID_s, C, \rho, t, R, a_0, a_1, \ldots, a_{u-1})$, then sends it to $\mathcal{A}_1^1$.

(b) Otherwise, $\mathcal{C}$ randomly chooses $r, sk, \{k_j\}_{j=1}^u, t, d_1, d_2, d_3 \in Z_q^*$ and $\zeta \in \{0,1\}^{2|G|+|m|}$. Next, $\mathcal{C}$ calculates $b = batVal \bmod rsk$, $R = rP$, $\omega_j = (r+b)IPK_j, (j = 1, 2, \ldots, u)$, constructs a polynomial $f(x) = \prod_{j=1}^u (x - k_j) + SK = a_0 + a_1x^1 + a_2x^2 + \cdots + a_{u-1}x^{u-1} + x^u$, and calculates the ciphertext $C = \zeta \oplus (PK_s||Y_s||m)$, and $\rho = r + d_1y_s + d_2sk_s + d_3b$. Then, it updates L_{H3}, L_{H4}, and L_{H5}. Finally, $\mathcal{C}$ sends $\varepsilon = (PID_s, C, \rho, t, R, a_0, a_1, \ldots, a_{u-1})$ to $\mathcal{A}_1^1$.

mulDeSignCryptQuery: Upon receiving $\varepsilon = (PID_j, C, \rho, t, R, a_0, a_1, \ldots, a_{u-1})$ to $\mathcal{A}_1^1$, $\mathcal{C}$ obtains $h_j, r, sk, \{k_j\}_{j=1}^u, t, d_1, d_2, d_3 \in Z_q^*$ and $\zeta \in \{0,1\}^{2|G|+|m|}$ from L_{H2}, L_{H3}, L_{H4}, and L_{H5}. And $\mathcal{C}$ calculates $\{\omega_j = sk_j(R + B)\}_{j=1}^u$, and constructs a polynomial $f(x) = a_0 + a_1x^1 + a_2x^2 + \cdots + a_{u-1}x^{u-1} + x^u$ using the received polynomial coefficients: $a_0, a_1, \ldots, a_{u-1}$. Then $\mathcal{C}$ calculates the session key $SK = f(k_j)$, $(PK_s, Y_s, m) = \zeta \oplus C$. Finally, $\mathcal{C}$ sends (PK_s, Y_s, m) to $\mathcal{A}_1^1$.

Challenge: After the above queries, $\mathcal{A}_1^1$ generates $m_0, m_1, PIDs$, and $REV = \{IPK_j\}_{j=1}^u$, then sends them to $\mathcal{C}$, where m_0 and m_1 are two messages of different content and equal length. Next, $\mathcal{C}$ selects $\beta \in \{0,1\}$. $\mathcal{C}$ randomly chooses $r, sk, \{k_j\}_{j=1}^u, t, d_1, d_2, d_3 \in Z_q^*$ and $\zeta \in \{0,1\}^{2|G|+|m|}$. Next, $\mathcal{C}$ calculates $b = batVal \bmod rsk$, $R = rP$, $\{\omega_j = (r+b)IPK_j\}_{j=1}^u$, constructs a polynomial $f(x) = \prod_{j=1}^u (x - k_j) + SK = a_0 + a_1x^1 + a_2x^2 + \cdots + a_{u-1}x^{u-1} +$

x^u, and calculates the ciphertext $C = \zeta \oplus (PK_s||Y_s||m_\beta)$, and $\rho = r + d_1 y_s + d_2 sk_s + d_3 b$. Then, it stores them in L_{H3}, L_{H4}, and L_{H5}, respectively. Finally, $\mathcal{C}$ sends $\varepsilon^* = (PID_s, C, \rho, t, R, a_0, a_1, \ldots, a_{u-1})$ to $\mathcal{A}_1^1$.

Phase 2: $\mathcal{A}_1^1$ can execute the same series of above queries except for $mulDeSignCryptQuery(\varepsilon^*)$.

Guess: $\mathcal{A}_1^1$ guesses a bit $\beta \in \{0,1\}$ and wins the game if $\beta = \beta'$.

If $\mathcal{A}_1^1$ wants to succeed, it must obtain correct keys $\{k_j\}_{j=1}^u$ and the corresponding $\{\omega_j\}_{j=1}^u$. Then, $\mathcal{C}$ outputs the solution of CDHP: $\kappa\tau P = (d_1 h_j)^{-1}\rho PK_j + (\kappa\rho - \omega_j)(d_1 d_2)^{-1}P - (\kappa + sk_j)((h_j)^{-1}Y_j + (d_1 h_j)^{-1} d_2 PK_j) - (\kappa + sk_j)(1 - batVal)b(d_2 h_j)^{-1}$. Thus, $\mathcal{C}$ can solve CDHP with an advantage $\delta = \delta'$ in time $t' < t + O(Q_{pk} + Q_s + Q_u)$. □

Theorem 2. *Assume that $\mathcal{A}_2^1$ can win Game I with non-negligible advantage δ, then $\mathcal{C}$ can solve CDHP with non-negligible advantage δ'.*

Proof of Theorem 2. Suppose challenger $\mathcal{C}$ obtains an instance of the CDHP problem $(P, \kappa P, \tau P)$ and attempts to compute $\kappa\tau P$. $\mathcal{C}$ and $\mathcal{A}_2^1$ can perform the same interactions as Theorem 1. In this game $\mathcal{A}_2^1$ knows $msk = \kappa$, but it cannot initiate *replacePublicKeyQuery*.

Guess: $\mathcal{A}_2^1$ guesses a bit $\beta \in \{0,1\}$ and wins the game if $\beta = \beta'$.

If $\mathcal{A}_2^1$ wants to succeed, it must obtain correct keys $\{k_j\}_{j=1}^u$ and the corresponding $\{\omega_j\}_{j=1}^u$. Then, $\mathcal{C}$ outputs the solution of CDHP: $\kappa\tau P = (\kappa\rho P - \omega_j P - \kappa d_1 Y_j - d_2 PK_j + (1 - d_3 B))(d_1)^{-1}$. Thus, $\mathcal{C}$ can solve CDHP with an advantage $\delta = \delta'$ in time $t' < t + O(Q_{pk} + Q_s + Q_u)$. □

6.2.2. Unforgeability

Our scheme can achieve unforgeability, which is shown in Theorems 3 and 4.

Theorem 3. *Assume that $\mathcal{A}_1^2$ can win Game III with non-negligible advantage δ, then $\mathcal{C}$ can solve CDHP with non-negligible advantage δ'.*

Proof of Theorem 3. Suppose challenger $\mathcal{C}$ obtains a CDHP instance $(P, \kappa P, \tau P)$ and attempts to compute $\kappa\tau P$. $\mathcal{C}$ and $\mathcal{A}_1^2$ can perform the same interactions as stage 1 and stage 2 in Theorem 1.

Forgery: $\mathcal{A}_1^2$ outputs a forgery $\varepsilon^{*'} = (PID_s^*, C^*, \rho^*, t^*, R^*, a_0, a_1, \ldots, a_{u-1})$ with PID_s^*. And $\mathcal{A}_1^2$ cannot query the secret key on PID_s^*. If the equation $R^* = \rho^* P - d_1 Y_j^* - d_1 h_j^* bP - d_2(sk_j + \kappa)P - d_3 B$ holds, $\mathcal{A}_1^2$ will succeed. Finally, $\mathcal{C}$ receives the forged signature $\varepsilon^{*'} = (PID_s^*, C^*, \rho^* = r^* + d_1 y^* + d_2(sk_j + \kappa) + d_3 b, t^*, R^*, a_0, a_1, \ldots, a_{u-1})$.

After the relevant queries are executed, the corresponding public key with the same identity will be changed. Therefore, $\mathcal{C}$ can forge a signature with an advantage $\delta' = \delta$ in time $t' < t + O(Q_{pk} + Q_s + Q_u)$. □

Theorem 4. *Assume that $\mathcal{A}_2^2$ can win Game IV with non-negligible advantage δ, then $\mathcal{C}$ can solve CDHP with non-negligible advantage δ'.*

Proof of Theorem 4. Assume that challenger $\mathcal{C}$ obtains a CDHP instance $(P, \kappa P, \tau P)$ and attempts to compute $\kappa\tau P$. $\mathcal{C}$ and $\mathcal{A}_2^2$ can perform the same interactions as stage 1 and stage 2 in Theorem 2.

Forgery: $\mathcal{A}_2^2$ outputs a forgery $\varepsilon^{*'} = (PID_s^*, C^*, \rho^*, t^*, R^*, a_0, a_1, \ldots, a_{u-1})$ with PID_s^*. And $\mathcal{A}_2^2$ cannot query the secret key on PID_s^*. If the equation $R^* = \rho^* P - d_1 Y_j^* - d_1 h_j^* \kappa P - d_2(sk_j + \tau)P - d_3 B$ holds, $\mathcal{A}_2^2$ will succeed. Finally, $\mathcal{C}$ receives the forged signature $\varepsilon^{*'} = (PID_s^*, C^*, \rho^* = r^* + d_1 y^* + d_2(sk_j + \tau) + d_3 b, t^*, R^*, a_0, a_1, \ldots, a_{u-1})$.

After the relevant queries are executed, the corresponding public key with the same identity will be changed. Therefore, $\mathcal{C}$ can forge a signature with an advantage $\delta' = \delta$ in time $t' < t + O(Q_{pk} + Q_s + Q_u)$. □

7. Performance Evaluation

In this section, we evaluate the performance of our URSCS scheme, and then compare computation overheads and communication overheads of our URSCS with those of the state-of-the-art provably secure signcryption schemes, specifically Yang et al., 2022 [8], Yang et al., 2022 [13], Deng 2020, [17], Liang et al., 2023 [18]. Since the registration phase is a one-time process in the scenario of VANETs, the computation and communication overheads can be neglected; therefore, we focus on the signcryption phase (the signcryption of messages during one-to-many communication) and the unsigncryption phase.

7.1. Computation Overhead

To calculate the computational overhead, we conducted simulations using a pairing-based cryptography library and a multi-precision integer and rational arithmetic C/C++ library. The execution times for common cryptographic operations were obtained by averaging multiple execution times with different input values. We denote the execution time of a modular exponentiation operation as T_{mod}, the execution time of a scale multiplication on ECC as T_{me}, the execution time of a point addition on ECC as Tae, the execution time of a bilinear mapping operation as T_{bm}, the execution time of a scale multiplication on bilinear pairing as T_{mbm}, and the execution time of a hash operation as T_h. Additionally, the execution times of those operations on a desktop equipped with an 11th Gen Intel(R) Core(TM) i7-11800H processor operating at 2.30 GHz and 16.0 GB of RAM are summarized in Table 3. The execution times of other operations, such as hashing, are so small that they can be neglected compared to the operations mentioned above; therefore, their times are not considered in our experiments.

Table 3. Execution time of basic operations.

Notation	Description	Run Time (ms)
T_m	Modular exponentiation	0.224
T_{me}	Scale multiplication on ECC	0.518
T_{ae}	Point addition on ECC	0.018
T_{bm}	Bilinear mapping operation	5.917
T_{mbm}	Scale multiplication on bilinear pairing	1.919
T_h	Hash operation	0.002

In our URSCS scheme, the signcryption phase requires $u + 1$ scalar multiplications operations on ECC, $u + 4$ hash operations, where u represents the number of receivers, so the signcryption cost of our scheme is $(u + 1)T_{me} + (u + 4)T_h$. Similarly, the unsigncryption phase needs $5u$ point addition operations on ECC, $6u$ scalar multiplications operations on ECC, $6u$ hash operations. Thus, the unsigncryption cost of our scheme is $5uT_{ae} + 6uT_{me} + 6uT_h$, and the total computation cost of our scheme can be expressed as $(7u + 1)T_{me} + (7u + 4)T_h + 5uT_{ae}$.

In scheme Yang et al., 2022 [8], the signcryption needs $6u$ scalar multiplication operations on ECC, $4u$ point addition operations on ECC, and $5u$ hash operations, so the signcryption cost is $6uT_{me} + 4uT_{ae} + 5uT_h$. Similarly, the unsigncryption needs $3u$ scalar multiplication operations on ECC, $5u - 4$ point addition operations on ECC, $4u + 3$ hash operations, and 5 bilinear pairing operations, so the unsigncryption cost is $3uT_{me} + (5u - 4)T_{ae} + (4u + 3)T_h + 5T_{bm}$, and the total computation cost of scheme Yang et al., 2022 [8] is $9uT_{me} + (9u - 4)T_{ae} + (9u + 3)T_h + 5T_{bm}$.

In scheme Yang et al., 2022 [13], the signcryption phase needs $3u$ scalar multiplication operations on ECC, u point addition operations on ECC, u modular exponentiation operations, and $3u$ hash operations, so the signcryption cost is $3uT_{me} + uT_{ae} + uT_m + 3uT_h$.

Similarly, the unsigncryption phase needs $3u$ scalar multiplication operations on ECC, $3u$ point addition operations on ECC, $3u$ hash operations, and u bilinear pairing operations, so the unsigncryption cost is $3uT_{me} + 3uT_{ae} + 3uT_h + uT_{bm}$. Then, the total computation cost of scheme Yang et al., 2022 [13] is $6uT_{me} + 4uT_{ae} + uT_m + 6uT_h + uT_{bm}$.

In scheme Deng 2020, [17], the signcryption phase needs $2u + 3$ scalar multiplication operations on ECC and u bilinear pairing operations, so the signcryption cost is $(2u + 3)T_{me} + uT_{bm}$. Similarly, the unsigncryption phase needs one bilinear pairing operation and one modular exponentiation operation, so the unsigncryption cost is $T_{bm} + T_m$. Then, the total computation cost of scheme Deng 2020, [17] is $(u + 1)T_{bm} + (2u + 3)T_{me} + T_m$.

In scheme Liang et al., 2023 [18], the signcryption phase needs $u + 1$ scalar multiplication operations on ECC, $u + 4$ hash operations. So the signcryption cost is $(u + 1)T_{me} + (u + 4)T_h$. Similarly, the unsigncryption phase needs $6u$ scalar multiplication operations on ECC, $5u$ point addition operations on ECC, and $6u$ hash operations, so the unsigncryption cost is $6uT_{me} + 5uT_{ae} + 6uT_h$. Then, the total computation cost of schemeLiang et al., 2023 [18] is $(7u + 1)T_{me} + (7u + 4)T_h + 5uT_{ae}$.

Also, Table 4 shows the total computation cost of each algorithm, respectively. The experimental comparison is depicted in Figure 2, where the number of receivers has increased from 0 to 30. This figure evidently shows that our scheme has apparent advantages over schemes Yang et al., 2022 [8], Yang et al., 2022 [13], Deng 2020, [17]. The efficiency of scheme Liang et al., 2023 [18] is similar to that of our scheme, but its receivers are limited to RSUs, while the receivers of our scheme can be both RSUs and vehicles, with greater flexibility.

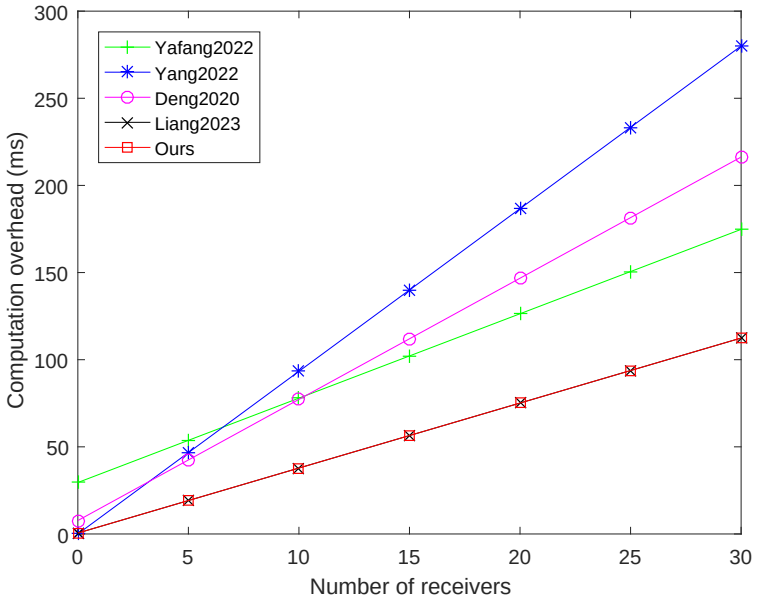

Figure 2. Comparison of computation overhead for different number of receivers [8,13,17,18].

Table 4. Comparison of computation overhead.

Scheme	Signcryption (ms)	UnSigncryption (ms)	Total Cost (ms)
Ours	$(u+1)T_{me} + (u+4)T_h$	$5uT_{ae} + 6uT_{me} + 6uT_h$	$(7u+1)T_{me} + (7u+4)T_h + 5uT_{ae}$
Yang et al., 2022 [8]	$6uT_{me} + 4uT_{ae} + 5uT_h$	$3uT_{me} + (5u-4)T_{ae} + (4u+3)T_h + 5T_{bm}$	$9uT_{me} + (9u-4)T_{ae} + (9u+3)T_h + 5T_{bm}$
Yang et al., 2022 [13]	$3uT_{me} + uT_{ae} + uT_m + 3uT_h$	$3uT_{me} + 3uT_{ae} + 3uT_h + uT_{bm}$	$6uT_{me} + 4uT_{ae} + uT_m + 6uT_h + uT_{bm}$
Deng 2020, [17]	$(2u+3)T_{me} + uT_{bm}$	$T_{bm} + T_m$	$(u+1)T_{bm} + (2u+3)T_{me} + T_m$
Liang et al., 2023 [18]	$(u+1)T_{me} + (u+4)T_h$	$6uT_{me} + 5uT_{ae} + 6uT_h$	$(7u+1)T_{me} + (7u+4)T_h + 5uT_{ae}$

7.2. Communication Overhead

In this section, given a secure length of 128 bits [13], we compared the communication overhead of our URSCS scheme with those of the state-of-the-art provably secure signcryption schemes, specifically Yang et al., 2022 [8], Yang et al., 2022 [13], Deng 2020, [17], Liang et al., 2023 [18]. In schemes based on bilinear pairing, the size of the element in G_T is 128 bytes, namely $|G_T| = 128$ bytes; while in schemes based on ECC, the size of the elements in G is 64 bytes, namely $|G| = 64$ bytes. The size of the elements in Z_q^*, real identity, message, the timestamp are 32 bytes, 32 bytes, 32 bytes, 4 bytes, respectively, namely $\left|Z_q^*\right| = 32$ bytes, $|RID| = 32$ bytes, $|m| = 32$ bytes, $|t| = 4$ bytes.

In our URSCS scheme, during the processes of signcryption and unsigncryption, the information that needs to be transmitted is $\varepsilon = (PID_s, C, \rho, t, R, a_0, a_1, \ldots, a_{u-1})$, where the size of pseudonym $PID_s = (PID_s^1 = x_s P, PID_s^2 = RID \oplus H_1(mskPID_s^1), t_s^{pid})$ is $(64 + 32 + 4) = 100$ bytes, the size of ciphertext $C = H_4(SK, PID_s) \oplus (PK_s||Y_s||m)$ is $(64 + 64 + 32) = 160$ bytes, and u indicates the number of receivers. So the total communication overhead is $(64 + 32 + 4) + (64 + 64 + 32) + 32 + 4 + 64 + 32u = 360 + 32u$ bytes.

We utilized the same method to compute the communication overheads of the schemes [8,13,17,18], where the results are illustrated in Figure 3 and Table 5. Compared to other schemes, our approach has lower communication overhead. Although the communication overhead of the scheme Liang et al., 2023 [18] is lower than that of our scheme, but it did not consider the issue of elements needing to be coprime in the CRT, and it also has the limitation that messages can only be sent to RSUs. Similarly, although scheme Deng 2020, [17] has lower communication overhead, its computational cost is too high, and it lacks tracking and revocation functions.

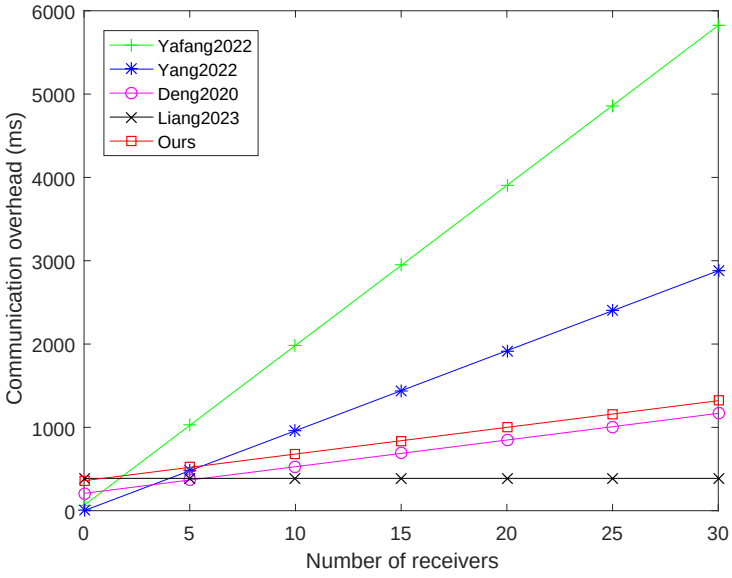

Figure 3. Comparison of communication overhead for different number of receivers [8,13,17,18].

Table 5. Comparison of communication overhead.

Scheme	Communication Overhead (bytes)
Ours	$(u+1)\left\|Z_q^*\right\| + 4\|G\| + \|m\| + 2\|t\| + \|RID\| = 360 + 32u$
Yang et al., 2022 [8]	$(2u+1)\|G\| + u(Z_q^* + \|m\|) = 64 + 192u$
Yang et al., 2022 [13]	$u\|G\| + u\|m\| = 96u$
Deng 2020, [17]	$(u+1)\left\|Z_q^*\right\| + 2\|G\| + \|C\| = 208 + 32u$
Liang et al., 2023 [18]	$2\left\|Z_q^*\right\| + 4\|G\| + \|m\| + \|t\| + \|RID\| = 388$

7.3. Comparison of Revocation Public Key Transmission Overhead

Scheme Liang et al., 2023 [18] is the most outstanding unlinkable multi-recipient scheme with revocation mechanism so far. However, the implementation of its revocation key mechanism requires the construction of a complex polynomial. The number of coefficients of the polynomial is the same as the number of vehicles. The master revocation public key parameters to be transmitted are $\{B, a_0, a_1, a_2, \ldots, a_{count-1}\}$, where B is a point on the discrete elliptic curve group, and $a_0, a_1, a_2, \ldots, a_{count-1} \in Z_q^*$, which are the coefficients of the polynomial. Therefore, the communication overhead for transmitting the master revocation public key parameters is $64 + 32count$ bytes. When there are a large number of vehicles within the domain of a TCC, such as 100,000 vehicles, the master revocation public key parameters generated at one time will include 100,001 numbers. On the other hand, the update of the master revocation public key is relatively frequent. When a vehicle joins the TCC's domain or a vehicle needs to be revoked, the master revocation secret key and master revocation public key need to be updated again. Thus, the overhead of scheme Liang et al., 2023 [18] in transmitting the master revocation public key is significant.

To reduce the communication overhead of transmitting the revocation public key, we present a new revocation key mechanism based on CRT. The revocation public key parameters to be transmitted are $(B, batVal)$, where B is a point on the discrete elliptic curve group, and $batVal \in Z_q^*$. Regardless of the number of vehicles within the TCC domain, the communication overhead for transmitting the master revocation public key parameters is fixed, namely 96 bytes $(64 + 32)$. Thus, our URSCS scheme has an absolute advantage in the transmission overhead of the revocation public key. Figure 4 demonstrates the apparent advantage of our scheme in transmission overhead of transmitting the revocation public key.

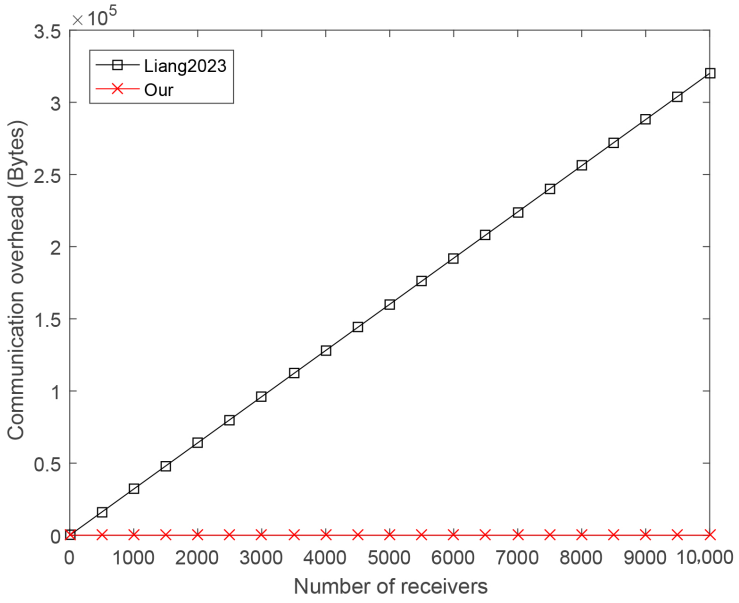

Figure 4. Comparison of revocation public key transmission overhead [18].

8. Conclusions

Based on the advantages of certificateless and signcryption, we developed a secure multi-receiver unlinkable communication scheme, allowing a sender to send the same signcryption information to multiple receivers, which can be either vehicles or RSUs. Through anonymous transmission and the instant update of key pairs when sending messages, our scheme ensures that malicious attackers cannot trace the sender through multiple messages originating from the same source, thereby achieving the unlinkability of the sender. Moreover, this paper also constructed a revocation key mechanism with low communication cost by utilizing CRT. Furthermore, the comprehensive security analysis demonstrated that

our scheme exhibits superior security performance. In comparison with the state-of-the-art schemes, our scheme has low communication overhead and computation overhead. Note that this scheme solely focuses on a secure multi-receiver communication scheme under only one TCC, and the secure communication among various TCCs remains unexplored in this paper, which will be a key focus of our future research.

Author Contributions: Conceptualization, L.L. and Y.W.; methodology, L.L., Y.L. (Yangfan Liang) and D.C.; validation, Y.W. and Y.L. (Yangfan Liang); formal analysis, L.L. and D.C.; Software, X.W.; writing—original draft preparation, L.L. and D.C.; writing—review and editing, L.L., Y.W., X.W. and Y.L. (Yining Liu); visualization, D.C.; project administration, Y.L. (Yining Liu). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original data presented in the study are openly available in [Pairing-Based Cryptography Library] at [https://crypto.stanford.edu/pbc] and [Miracl Library] at [https://github.com/miracl/MIRACL], both accessed on 9 August 2024.

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

References

1. Chowdhury, D.N.; Agarwal, N.; Laha, A.B.; Mukherjee, A. A vehicle-to-vehicle communication system using Iot approach. In Proceedings of the 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 29–31 March 2018; IEEE: New York, NY, USA, 2018; pp. 915–919.
2. Zhang, J.; Jiang, Y.; Cui, J.; He, D.; Bolodurina, I.; Zhong, H. DBCPA: Dual Blockchain-Assisted Conditional Privacy-Preserving Authentication Framework and Protocol for Vehicular Ad Hoc Networks. *IEEE Trans. Mob. Comput.* **2024**, *23*, 1127–1141. [CrossRef]
3. Al-shareeda, M.A.; Alazzawi, M.A.; Anbar, M.; Manickam, S.; Al-Ani, A.K. A comprehensive survey on vehicular ad hoc networks (vanets). In Proceedings of the 2021 International Conference on Advanced Computer Applications (ACA), Maysan, Iraq, 25–26 July 2021; IEEE: New York, NY, USA, 2021; pp. 156–160.
4. Liang, Y.; Liu, Y. Analysis and improvement of an efficient certificateless aggregate signature with conditional privacy preservation in VANETs. *IEEE Syst. J.* **2022**, *17*, 664–672. [CrossRef]
5. Biswas, M.; Das, D.; Banerjee, S.; Mukherjee, A.; AL-Numay, W.; Biswas, U.; Zhang, Y. Blockchain-Enabled Communication Framework for Secure and Trustworthy Internet of Vehicles. *Sustainability* **2023**, *15*, 9399. [CrossRef]
6. Liu, Z.; Wan, L.; Guo, J.; Huang, F.; Feng, X.; Wang, L.; Ma, J. PPRU: A Privacy-Preserving Reputation Updating Scheme for Cloud-Assisted Vehicular Networks. *IEEE Trans. Veh. Technol.* **2023**, 1–16. [CrossRef]
7. Xie, Q.; Ding, Z.; Xie, Q.; Tan, X.; He, D.; Tang, W. Blockchain-Based Traffic Accident Handling Protocol without Third-Party for VANETs. *IEEE Internet Things J.* **2024**, 1. [CrossRef]
8. 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]
9. Cao, L.; Ge, W. Analysis of Certificateless Signcryption Schemes and Construction of a Secure and Efficient Pairing-free one based on ECC. *KSII Trans. Internet Inf. Syst. (TIIS)* **2018**, *12*, 4527–4547.
10. Li, Y.; Qi, Y.; Lu, L. Secure and efficient V2V communications for heterogeneous vehicle ad hoc networks. In Proceedings of the 2017 International Conference on Networking and Network Applications (NaNA), Kathmandu City, Nepal, 16–19 October 2017; IEEE: New York, NY, USA, 2017; pp. 93–99.
11. Ali, I.; Lawrence, T.; Omala, A.A.; Li, F. An efficient hybrid signcryption scheme with conditional privacy-preservation for heterogeneous vehicular communication in VANETs. *IEEE Trans. Veh. Technol.* **2020**, *69*, 11266–11280. [CrossRef]
12. Abouelkheir, E.; El-sherbiny, S. Pairing free identity based aggregate signcryption scheme. *IET Inf. Secur.* **2020**, *14*, 625–632. [CrossRef]
13. Yang, Y.; He, D.; Vijayakumar, P.; Gupta, B.B.; Xie, Q. An efficient identity-based aggregate signcryption scheme with blockchain for IoT-enabled maritime transportation system. *IEEE Trans. Green Commun. Netw.* **2022**, *6*, 1520–1531. [CrossRef]
14. Wang, L.; Guan, Z.; Chen, Z.; Hu, M. Multi-receiver signcryption scheme with multiple key generation centers through public channel in edge computing. *China Commun.* **2022**, *19*, 177–198. [CrossRef]
15. Nkenyereye, L.; Liu, C.H.; Song, J. Towards secure and privacy preserving collision avoidance system in 5G fog based Internet of Vehicles. *Future Gener. Comput. Syst.* **2019**, *95*, 488–499. [CrossRef]
16. Ullah, I.; Khan, M.A.; Khan, F.; Jan, M.A.; Srinivasan, R.; Mastorakis, S.; Hussain, S.; Khattak, H. An efficient and secure multimessage and multireceiver signcryption scheme for edge-enabled internet of vehicles. *IEEE Internet Things J.* **2021**, *9*, 2688–2697. [CrossRef]
17. Deng, L. Anonymous certificateless multi-receiver encryption scheme for smart community management systems. *Soft Comput.* **2020**, *24*, 281–292. [CrossRef]

18. Liang, Y.; Yan, H.; Liu, Y. Unlinkable Signcryption Scheme for Multi-Receiver in VANETs. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 10138–10154. [CrossRef]
19. Wang, Y.; Wang, X.; Dai, H.N.; Zhang, X.; Imran, M. A Data Reporting Protocol With Revocable Anonymous Authentication for Edge-Assisted Intelligent Transport Systems. *IEEE Trans. Ind. Inform.* **2023**, *19*, 7835–7847. [CrossRef]
20. Azees, M.; Vijayakumar, P.; Deboarh, L.J. EAAP: Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 2467–2476. [CrossRef]
21. Li, J.; Lu, H.; Guizani, M. ACPN: A novel authentication framework with conditional privacy-preservation and non-repudiation for VANETs. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *26*, 938–948. [CrossRef]
22. Zhang, L.; Wu, Q.; Domingo-Ferrer, J.; Qin, B.; Hu, C. Distributed aggregate privacy-preserving authentication in VANETs. *IEEE Trans. Intell. Transp. Syst.* **2016**, *18*, 516–526. [CrossRef]
23. Zheng, Y. Digital signcryption or how to achieve cost (signature & encryption) significantly less than 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.
24. Zhang, A.; Wang, L.; Ye, X.; Lin, X. Light-weight and robust security-aware D2D-assist data transmission protocol for mobile-health systems. *IEEE Trans. Inf. Forensics Secur.* **2016**, *12*, 662–675. [CrossRef]
25. Zhou, C.X. An improved multi-receiver generalized signcryption scheme. *Int. J. Netw. Secur.* **2015**, *17*, 340–350.
26. 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.
27. Hellman, M. New directions in cryptography. *IEEE Trans. Inf. Theory* **1976**, *22*, 644–654.
28. Shamir, A. Identity-based cryptosystems and signature schemes. In Proceedings of the Advances in Cryptology: Proceedings of CRYPTO 84 4, Santa Barbara, CA, USA, 19–22 August 1984; Springer: Berlin/Heidelberg, Germany, 1985; pp. 47–53.
29. Barbosa, M.; Farshim, P. Certificateless signcryption. In Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security, Tokyo, Japan, 18–20 March 2008; Asia CCS '08; ACM: New York, NY, USA, 2008; pp. 369–372.
30. Wu, C.; Chen, Z. A new efficient certificateless signcryption scheme. In Proceedings of the 2008 International Symposium on Information Science and Engineering, Shanghai, China, 20–22 December 2008; IEEE: New York, NY, USA, 2008; Volume 1, pp. 661–664.
31. Sun, Y.; Li, H. ID-based signcryption KEM to multiple recipients. *Chin. J. Electron.* **2011**, *20*, 317–322.
32. Chen, J.; Wang, L.; Wen, M.; Zhang, K.; Chen, K. Efficient certificateless online/offline signcryption scheme for edge IoT devices. *IEEE Internet Things J.* **2021**, *9*, 8967–8979. [CrossRef]
33. Xie, W.; Zhang, Z. Efficient and provably secure certificateless signcryption from bilinear maps. In Proceedings of the 2010 IEEE International Conference on Wireless Communications, Networking and Information Security, Beijing, China, 25–27 June 2010; IEEE: New York, NY, USA, 2010; pp. 558–562.
34. Cui, M.; Han, D.; Wang, J. An efficient and safe road condition monitoring authentication scheme based on fog computing. *IEEE Internet Things J.* **2019**, *6*, 9076–9084. [CrossRef]
35. Xie, Z.; Chen, Y.; Ali, I.; Pan, C.; Li, F.; He, W. Efficient and Secure Certificateless Signcryption Without Pairing for Edge Computing-Based Internet of Vehicles. *IEEE Trans. Veh. Technol.* **2023**, *72*, 5642–5653. [CrossRef]
36. 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.* **2020**, *19*, 1464–1475. [CrossRef]
37. Yu, H.; Ren, R. Certificateless elliptic curve aggregate signcryption scheme. *IEEE Syst. J.* **2021**, *16*, 2347–2354. [CrossRef]
38. Pan, X.; Jin, Y.; Wang, Z.; Li, F. A pairing-free heterogeneous signcryption scheme for unmanned aerial vehicles. *IEEE Internet Things J.* **2022**, *9*, 19426–19437. [CrossRef]
39. Shim, K.A. CPAS: An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks. *IEEE Trans. Veh. Technol.* **2012**, *61*, 1874–1883. [CrossRef]
40. 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]
41. Zhu, F.; Yi, X.; Abuadbba, A.; Khalil, I.; Nepal, S.; Huang, X.; Yan, X. Certificate-based anonymous authentication with efficient aggregation for wireless medical sensor networks. *IEEE Internet Things J.* **2021**, *9*, 12209–12218. [CrossRef]
42. Qiao, Z.; Ma, K.; Zhou, Y.; Yang, Q.; Xia, Z.; Yang, B.; Zhang, M. An Anonymous and Efficient Certificate-Based Identity Authentication Protocol for VANET. *IEEE Internet Things J.* **2024**, *11*, 11232–11245. [CrossRef]
43. Gayathri, N.; Thumbur, G.; Reddy, P.V.; Ur Rahman, M.Z. Efficient Pairing-Free Certificateless Authentication Scheme with Batch Verification for Vehicular Ad-Hoc Networks. *IEEE Access* **2018**, *6*, 31808–31819. [CrossRef]
44. Zhou, Y.; Xu, R.; Qiao, Z.; Yang, B.; Xia, Z.; Zhang, M. An Anonymous and Efficient Multimessage and Multireceiver Certificateless Signcryption Scheme for VANET. *IEEE Internet Things J.* **2023**, *10*, 22823–22835. [CrossRef]