electronics

MDPI

Article

A Certificateless Online/Offline Aggregate Signcryption Scheme against Collusion Attacks Based on Fog Computing

Wanju Zhang [1], Shuanggen Liu [1,*], Yaowei Liu [1], Junjie Cao [2], Bingqi Fu [1] and Yun Du [1]

[1] School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China; zwj1247@stu.xupt.edu.cn (W.Z.); airlouise.liu@stu.xupt.edu.cn (Y.L.); fubingqi@stu.xupt.edu.cn (B.F.); duyundu@stu.xupt.edu.cn (Y.D.)

[2] PLA Rocket Force Engineering University, Xi'an 710025, China; cjjw0929@163.com

* Correspondence: liushuanggen201@xupt.edu.cn

Abstract: The certificateless online/offline aggregate signcryption scheme combines the characteristics of the certificateless aggregate signcryption scheme and the online/offline encryption scheme, which can increase efficiency while simultaneously reducing consumption. Some schemes can meet the requirements of confidentiality and real-time transmission of the data in ad hoc networks (VANETS). However, they are unable to withstand collusion attempts. A brand-new certificateless aggregate signcryption approach is suggested to overcome this problem. First, combining fog computing with online/offline encryption (OOE) technology can increase efficiency while simultaneously reducing consumption. Second, we may achieve effective information authentication and vehicle identification using aggregation and vehicle pseudonym systems. Third, the anti-collusion component is suggested as a viable defense against collusion assaults since certain methods are unable to withstand such attacks. Additionally, it is demonstrated that the technique has unforgeability and secrecy, and can fend off collusion attacks using the random oracle model. The findings demonstrate that our system can not only ensure the confidentiality and the real-time transmission of data but also resist collusion attacks without raising computational costs.

Keywords: vehicular ad hoc networks; fog computing; online/offline encryption; vehicle pseudonym system; anti-collusion factor

check for updates

Citation: Zhang, W.; Liu, S.; Liu, Y.; Cao, J.; Fu, B.; Du, Y. A Certificateless Online/Offline Aggregate Signcryption Scheme against Collusion Attacks Based on Fog Computing. *Electronics* **2023**, *12*, 4747. https://doi.org/10.3390/electronics12234747

Academic Editor: Andreas Mauthe

Received: 14 October 2023
Revised: 19 November 2023
Accepted: 20 November 2023
Published: 23 November 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the development of science and technology, autonomous driving technology has been implemented; however, further research is needed on how to achieve reliable autonomous driving in various weather and road conditions. In order to detect external information such as road conditions and provide better service for users, vehicular ad hoc networks (VANETs) have been created [1].

To better process these data in VANETS, Eric Schmidt, then Google's CEO, first proposed the concept of cloud computing at the Search Engine Conference in 2006. In the cloud architecture, as shown in Figure 1, all data are first sent to the cloud, processed by the cloud server, and then returned to the user. However, global mobile data are showing exponential growth, and some new applications have higher requirements for data transmission speed and efficiency, for example, autonomous driving. This has led to the gradual emergence of cloud computing issues based on cloud architectures. The process of transmitting data from sensing devices to cloud servers for data processing requires a long communication link, which can result in high latency. This characteristic is not suitable for VANETs that require high real-time data requirements.

Based on the above issue, Carnegie Mellon University proposed the concept of micro-clouds in 2009 [2], this can be said to be the embryonic form of fog computing. Cisco first proposed the concept of "fog computing" at the Cisco Live 2014 conference [3]. Fog computing is an extended concept of cloud computing; as shown in Figure 2, it is located between cloud servers and end users and provides them with services [4]. In the fog architecture, fog nodes are composed of a large number of weaker and more dispersed computable devices that can perform preliminary processing and analysis of data from users near the edge [5], and finally send the final results to the cloud center for long-term storage. In summary, the fog-based architecture supports mobility, location awareness, and low latency, which is suitable for VANETs.

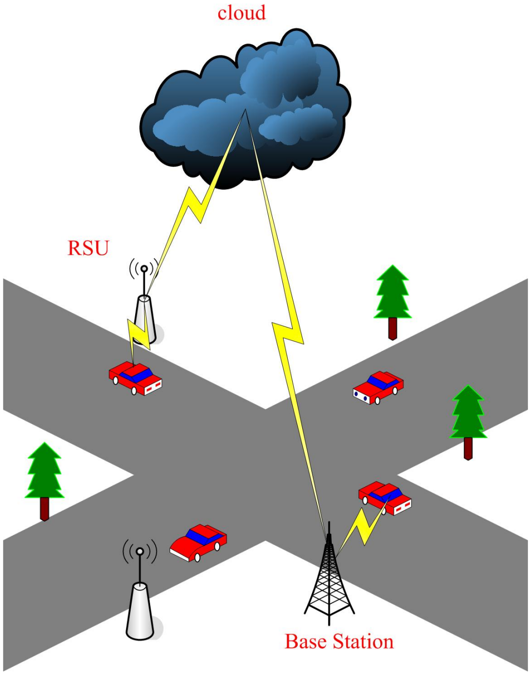


Figure 1. Cloud-based architecture.

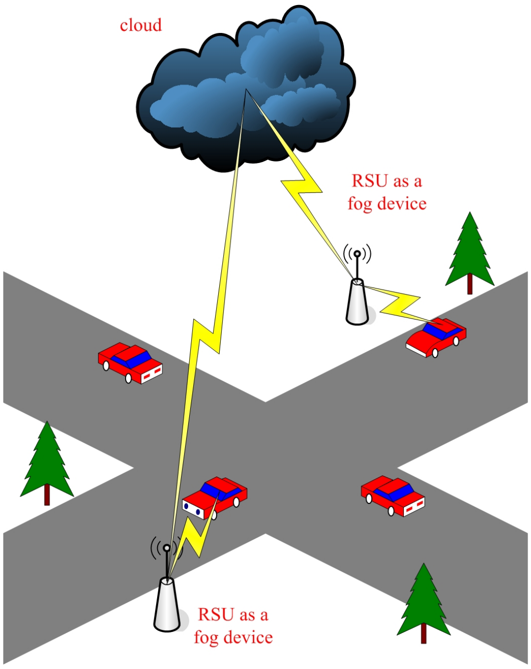


Figure 2. Fog-based architecture.

With the development of the Internet of Vehicles, there are not only increasingly improved technologies but also inevitable safety issues, and thus the safety issues of the

Internet of Vehicles have attracted more and more of people's attention [6]. For example, in the Internet of Vehicles, attackers can eavesdrop on public channels to obtain some identity information related to vehicle users or directly eavesdrop on sent messages for further attacks. Similarly, attackers can also intercept the information sent by vehicle users and tamper with it before sending it to the fog node. In this way, the fog node will feedback to the user with incorrect decisions, and in severe cases, it can also cause irreversible results. To avoid the aforementioned attacks, it is essential to sign, authenticate, and encrypt messages, as well as authenticate user identities during communication in the Internet of Vehicles. However, the massive amount of data in the Internet of Vehicles can result in significant computing and communication costs. Thus, building a more efficient and secure solution is well worth exploring.

In previous schemes, signature and encryption were separated, but over time, signcryption technology gradually evolved. It achieves both signature and encryption in one step, reducing costs while achieving confidentiality, integrity, authenticity, reliability, and nonrepudiation [7,8]. In recent years, there have been many algorithms based on public key infrastructure (PKI) cryptography that fall under public-key cryptography. However, due to the exponential growth of intelligent devices in the Internet of Vehicles, this signcryption algorithm can cause serious certificate management problems. Although identity-based cryptography (IDBC) algorithms solve the problem of digital certificate management in the Internet of Vehicles, the concern of key escrow still persists. To address the above issues, Al-Riyami et al. [9] proposed a new scheme. In this scheme, KGC only generates a portion of the user's public and private keys, and the complete public and private keys are generated by themselves. Even KGC cannot acquire the user's complete private key, which fundamentally solves the problem of key escrow.

The certificateless aggregate signcryption system (CLASC) and online/offline encryption (OOE) are presented in order to further increase the workpiece ratio. While aggregation technology can combine several ciphertexts into one for batch verification in CLASC, signcryption technology can accomplish the signature and encryption in one step, saving time consumption. Online/offline encryption enhances efficiency while further enhancing the security of the scheme. A better CLASC was suggested by Eslami et al. [10] and Basudan et al. [11], considerably enhancing security. The developers of the technique [12] proposed an identity-based online/offline scheme that drastically reduced computational costs.

1.1. Related Work

In this section, the advancement of the fog architecture is reviewed first, and then CLASC and OOE encryption technologies are introduced. Ultimately, we explain a prevalent attack in the Internet of Vehicles known as a collusion attack and how our system utilizes these technologies to defend against collusion attacks.

1.1.1. Fog Architecture

With the exponential growth of data, traditional methods of sending data to cloud servers for processing can no longer meet people's real-time demands. This problem has been effectively solved by the appearance of fog computing [13]. The framework of fog computing is made up of three layers, including the end user layer, cloud server layer, and fog node layer. The fog nodes are located between cloud servers and end users and provides them with services [4]. In the fog architecture, fog nodes are constituted of a large number of weaker and more dispersed computable devices that can perform preliminary processing and analysis of data from users near the edge, and finally send the final results to the cloud center for long-term storage [14]. Dastjerdi et al. [14] introduced the idea of fog computing, and represented the point of fog computing and its lurking applications in IoT. Fog computing can be applied in multiple fields of IoT, including smart cities, health care, intelligent traffic systems, and so on. Especially in VANETs, fog computing decreases the waiting time. Erskine et al. [15] combined fog computing with other algorithms, which enhanced the security and efficiency in VANETs. Furthermore, one of the top advantages of fog computing is that it has the capability to back sensor networks on a large scale. In summary, the fog-based architecture supports mobility, location awareness, and low latency, which is suitable for VANETs.

1.1.2. Scheme for Certificateless Aggregate Signcryption

Key escrow is no longer an issue because of the development of certificateless aggregate signcryption technology, which also resolves user public key distribution and digital certificate management issues [16]. Batch verification is accomplished by combining numerous ciphertexts using aggregation technology, which significantly increases the verification efficiency. A CLASC system was proposed by Lu and Xie [17], whilst a CASCF scheme was proposed by Kim et al. [18]. Nevertheless, both of them were inefficient since they needed bilinear pairings. Thus, it is imperative to design the CLASC scheme without bilinear pairings [19] in order to maximize efficiency. The authors in [16] throughly examined some of the most popular certificateless aggregation signcryption schemes in terms of computation and communication costs. From the scheme in [16], we can draw the conclusion that it is necessary to design a scheme without using bilinear pairings in the recent stage. To make sure of the safe and strong communication in the transportation networks, a pairing-free certificateless aggregate signcryption scheme was designed in [19]. Without using pairings, this scheme improved efficiency compared with other schemes.

1.1.3. Technology for Online and Offline Encryption

There are two stages in the OOE process: offline and online. It performs a large number of labor-intensive tasks while offline without knowing the encrypted message. However, it only performs light actions during the online phase. Online encryption will thus likely advance quickly. An online/offline heterogeneous signcryption technique that offers a workable solution for disparate mechanisms between CLC and PKI was presented by Hou et al. [20]. Additionally, a more effective and cost-effective online/offline encryption system based on identification with brief ciphertext was presented by Lai et al. [21]. In [22], a certificateless online/offline signcryption scheme proposed by An et al. was designed to ensure data unforgeability and confidentiality, enabling secure, lightweight data sharing in smart home systems. Compared with other schemes, it reduced communication costs as well as computation costs.

1.1.4. Collusion Attack

Collusion attacks often occur in the Internet of Vehicles, such as two attackers maliciously exchanging their signcryption information, which can be verified successfully. This can have the potential to be highly dangerous and even cause irreversible damage to the vehicles. Cui et al. [23] proposed an efficient and safe road condition monitoring scheme, which was very suitable for the Internet of Vehicles; however, it could not resist collusion attacks. Combined with Pan et al. [24] and Cui et al. [23], we propose a new certificateless online/offline aggregate signcryption scheme based on fog computing, which is not only suitable for the Internet of Vehicles but also can resist collusion attacks.

1.2. Research Contributions

Nowadays, some schemes cannot meet the requirement of real-time transmission of the data in VANETs or cannot resist collusion attacks [23]. In [23], the authors proposed a certificateless aggregate signcryption scheme combined with online/offline encryption to monitor the condition of the road, which made real-time transmission possible. First, by using fog computing, their scheme can well support low latency, and this is very important in VANETS. Second, their scheme combined the certificateless aggregate signcryption scheme with online/offline encryption, which not only enhanced the security of the scheme, but also reduced the cost of time at the same time. However, the proposed scheme in [23] cannot resist collusion attacks. The discovered disadvantage means that the vehicles may be dangerous in some conditions when attackers deliberately trade their signcryption information. To better overcome the above issue, our scheme is proposed. The details of the contribution of our scheme are listed below.

- We suggest the anti-collusion component to thwart this type of assault, which is known as a collusion attack, taking into account that in particular schemes two vehicles might deliberately trade their signcryption information. This exchange can be successfully validated. Compared with the existing the scheme in [23], our proposed scheme can effectively resist collusion attacks;
- Based on the proposed scheme, we use fog computing to design a certificateless online/offline aggregate signcryption, which enhances the security of data and increases efficiency in the VANETs. The scheme realizes mutual authentication, anonymity, undeniability, untraceability, and confidentiality;
- In VANETs, the authentication of vehicle identity and the privacy of the messages as well as vehicles are both important [25]. In our scheme, we not only protect the privacy of messages but also use a vehicle pseudonym system to secure the privacy of vehicles as well as realize vehicle identification.

1.3. Paper Structure

The remainder of the paper is constructed as below. To begin with, we introduce the system model, attack model, and design objectives in Section 2. In Section 3, we list some essential knowledge. Our scheme is described in Section 4. In Sections 5 and 6, the security analysis and performance analysis of our scheme are presented. Lastly, we present the conclusion of the whole paper.

2. System Model, Attack Model, and Design objectives

The following is a description of the proposed scheme's detailed system model, attack model, and design objectives.

2.1. System Model

Our scheme sets up a total of five entities in combination with the Internet of Vehicles scenario, e.g., intelligent devices, RSU as a fog device, the TA as the trust authority, KGC as the key generation center, and CS as the cloud server. The architecture of our scheme is shown in Figure 3.

- Intelligent devices are always embedded in vehicles to send signals, location, and road events and accept relevant information;
- The RSU is regarded as a fog device since it is closer than a cloud. It can process data in real-time while having less computational and storage capacity than the cloud;
- The system's cloud server is designated as CS. The data will be sent here for long-term storage following processing at the RSU. Fog devices offer mobility, location awareness, and low latency in contrast to transmitting all of the sensor's data straight to the cloud;
- The TA is a trusted authority, playing a crucial role in system initialization and vehicle pseudonym generation;
- KGC, a dependable organization, is in charge of setting up the system and producing a portion of the customers' private keys. In particular, it cannot access the private data of RSUs and sensors.

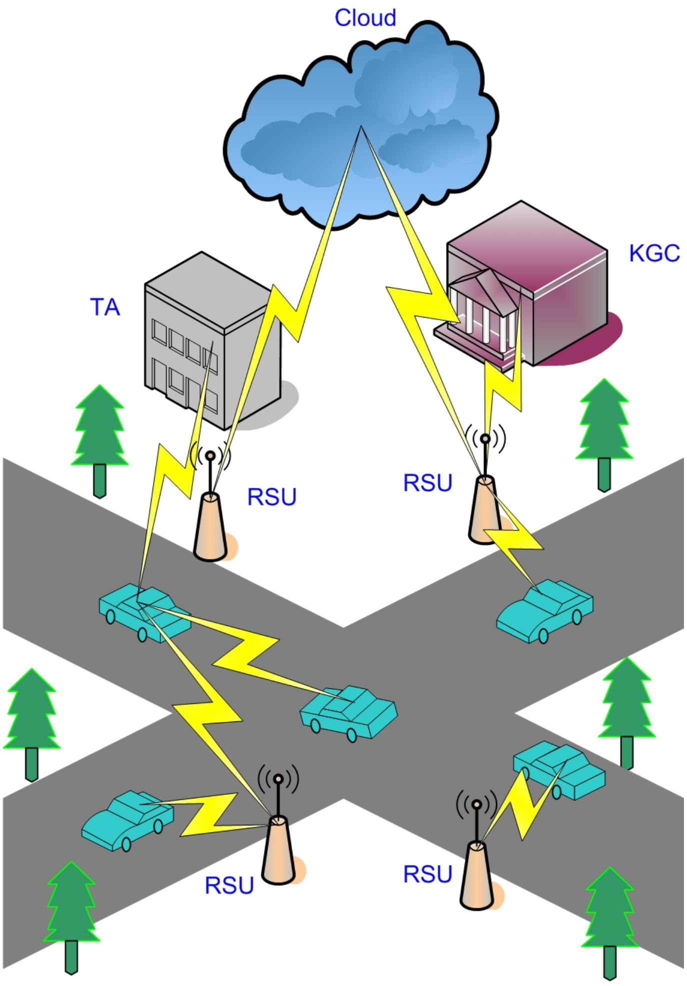


Figure 3. System model.

2.2. Attack Model

In this article, we make the assumption that RSU and user communication takes place over unsecured channels. It is clear that security risks exist for data transferred by the sensor or RSU. There are typically two different types of attacks in the LoV: passive attacks and active attacks.

Passive assaults, commonly referred to as eavesdropping, target a system's secrecy. The content of the communication can be obtained by two different passive assaults: message attacks and traffic analysis attacks.

- Attack obtaining the content of the message: Attackers can obtain user identity information or encrypted messages through eavesdropping on public channels or for further attacks;
- Traffic analysis attack: By observing and analyzing message patterns, the attackers can obtain the format of the message and determine the location and identity of both communication parties, which is sensitive to drivers.

Active attacks are initiated by attackers, including replay attacks, tampering attacks, and denial of service.

- Reply attack: The attacker sends the ciphertext that the RSU has received and verified successfully to deceive the system;
- Tampering attack: Attackers may modify the information originally sent, causing the RSU to receive incorrect information and make incorrect decisions;
- Denial of service: The attacker delivers the ciphertext that the RSU has successfully received and validated in a reply to trick the system.

2.3. Design Objectives

This study proposes a new collusion-resistant certificateless online/offline aggregate signcryption system based on fog computing. We want to fulfill the following requirements [26,27]:

- Mutual authentication: It is required between the RSU and the user in order to confirm the legitimacy of the network's members;
- Vehicle identity verification: By verifying the vehicle's identity through the TA, the sender of the message can be determined, enabling the tracking of malicious users to ensure the security of the vehicle's environment;
- The user anonymity: All users who send messages must be anonymous to ensure the privacy of their information;
- Untraceability: Although the enemy intercepted the communication message, they were unable to track the user's behavior;
- Undeniability: When a vehicle is involved in the authentication scheme, no vehicle can deny its behavior;
- Low computing cost: In the context of limited bandwidth, it is necessary to meet the requirement of low computing cost. Our proposed scheme is combines with CLASC and OOE, greatly reducing the computing overhead;
- Quick verification: Batch validation can be achieved through aggregation technology;
- Resist collusion attacks: In part of the schemes, two vehicles can maliciously exchange their signcryption information, which can be verified successfully. This can be very dangerous and even cause irreversible damage to the vehicles in the vehicular ad hoc networks (VANETs). Thus, it is very important to design a scheme that can resist collusion attacks.

3. Preliminaries

We provide some essential complexity assumptions in this section.

- The elliptic curve discrete logarithmic problem (also known as the elliptic curve discrete logarithm problem, or ECDLP): Given any point $Q \in G$, an $a \in Zq^*$ on the elliptic curve, and a set of elements (P, aP), solving a is challenging. G is an additive group on the elliptic curve of q-order and P is the generator of G;
- The elliptic curve computational Diffie–Hellman problem (also known as the EC-CDH assumption) is defined as follows: G is an additive group on the elliptic curve of q-order. Let P be one of G's generators. Given aP, $bP \in G$, and a, $b \in Zq^*$, it is difficult to compute abP;
- Hash collision resistance: Finding x and x' such that $H(x) = H(x')$ is a challenging task for a hash function.

4. Proposed Scheme

In this section, a secure scheme is proposed. The scheme process diagram is shown in Figure 4, and the description of symbols is shown in Table 1.

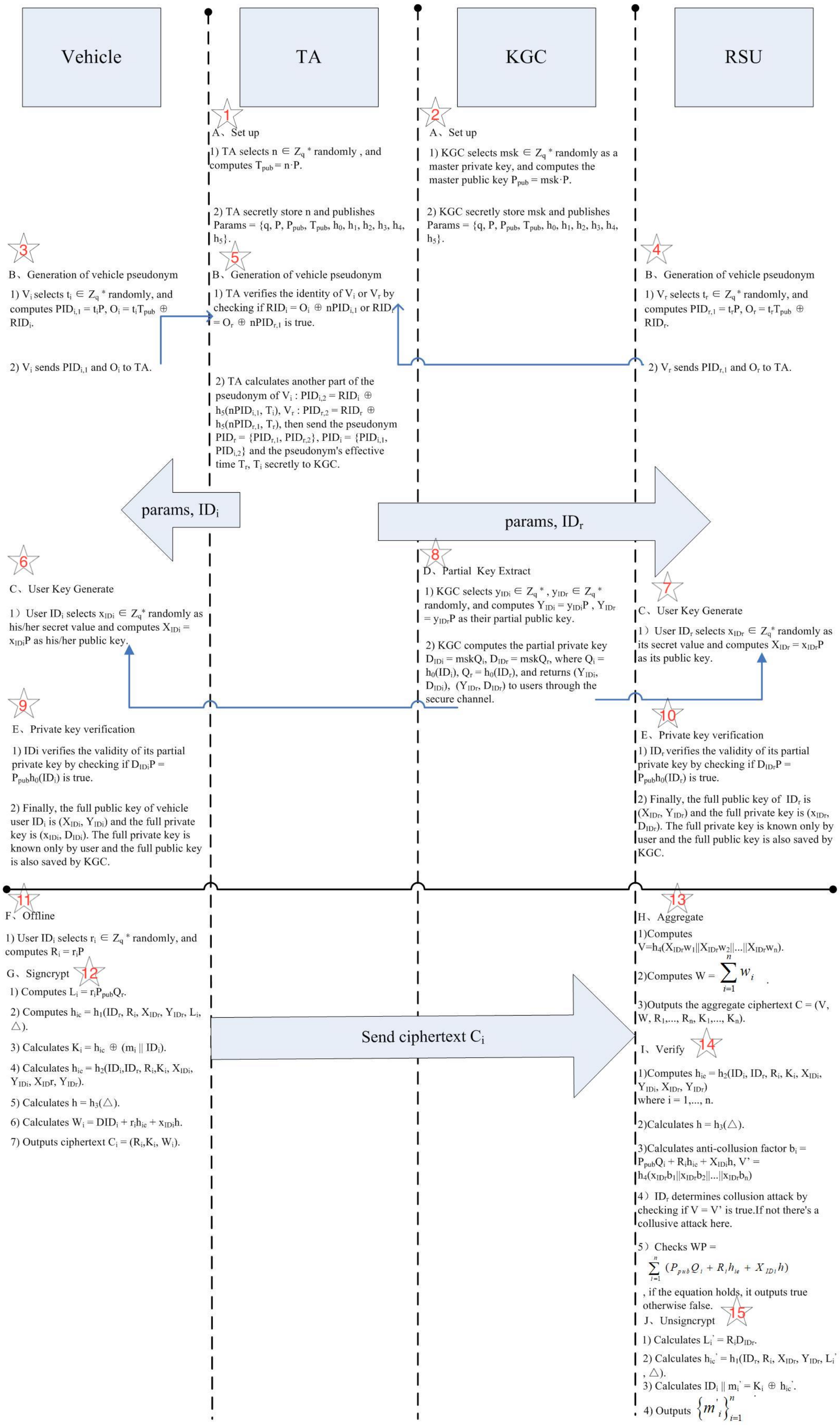

Vehicle
TA
KGC
RSU
1
A、Set up
1) TA selects $n \in Z_q{}^*$ randomly , and computes $T_{pub} = n \cdot P$.
2) TA secretly store n and publishes Params = $\{q, P, P_{pub}, T_{pub}, h_0, h_1, h_2, h_3, h_4, h_5\}$.
2
A、Set up
1) KGC selects $msk \in Z_q{}^*$ randomly as a master private key, and computes the master public key $P_{pub} = msk \cdot P$.
2) KGC secretly store msk and publishes Params = $\{q, P, P_{pub}, T_{pub}, h_0, h_1, h_2, h_3, h_4, h_5\}$.
3
B、Generation of vehicle pseudonym
1) V_i selects $t_i \in Z_q{}^*$ randomly, and computes $PID_{i,1} = t_iP$, $O_i = t_iT_{pub} \oplus RID_i$.
2) V_i sends $PID_{i,1}$ and O_i to TA.
5
B、Generation of vehicle pseudonym
1) TA verifies the identity of V_i or V_r by checking if $RID_i = O_i \oplus nPID_{i,1}$ or $RID_r = O_r \oplus nPID_{r,1}$ is true.
2) TA calculates another part of the pseudonym of V_i : $PID_{i,2} = RID_i \oplus h_5(nPID_{i,1}, T_i)$, V_r : $PID_{r,2} = RID_r \oplus h_5(nPID_{r,1}, T_r)$, then send the pseudonym $PID_r = \{PID_{r,1}, PID_{r,2}\}$, $PID_i = \{PID_{i,1}, PID_{i,2}\}$ and the pseudonym's effective time T_r, T_i secretly to KGC.
4
B、Generation of vehicle pseudonym
1) V_r selects $t_r \in Z_q{}^*$ randomly, and computes $PID_{r,1} = t_rP$, $O_r = t_rT_{pub} \oplus RID_r$.
2) V_r sends $PID_{r,1}$ and O_r to TA.
params, ID_i
params, ID_r
6
C、User Key Generate
1) User ID_i selects $x_{IDi} \in Z_q{}^*$ randomly as his/her secret value and computes $X_{IDi} = x_{IDi}P$ as his/her public key.
8
D、Partial Key Extract
1) KGC selects $y_{IDi} \in Z_q{}^*$, $y_{IDr} \in Z_q{}^*$ randomly, and computes $Y_{IDi} = y_{IDi}P$, $Y_{IDr} = y_{IDr}P$ as their partial public key.
2) KGC computes the partial private key $D_{IDi} = mskQ_i$, $D_{IDr} = mskQ_r$, where $Q_i = h_0(ID_i)$, $Q_r = h_0(ID_r)$, and returns (Y_{IDi}, D_{IDi}), (Y_{IDr}, D_{IDr}) to users through the secure channel.
7
C、User Key Generate
1) User ID_r selects $x_{IDr} \in Z_q{}^*$ randomly as its secret value and computes $X_{IDr} = x_{IDr}P$ as its public key.
9
E、Private key verification
1) IDi verifies the validity of its partial private key by checking if $D_{IDi}P = P_{pub}h_0(ID_i)$ is true.
2) Finally, the full public key of vehicle user ID_i is (X_{IDi}, Y_{IDi}) and the full private key is (x_{IDi}, D_{IDi}). The full private key is known only by user and the full public key is also saved by KGC.
10
E、Private key verification
1) ID_r verifies the validity of its partial private key by checking if $D_{IDr}P = P_{pub}h_0(ID_r)$ is true.
2) Finally, the full public key of ID_r is (X_{IDr}, Y_{IDr}) and the full private key is (x_{IDr}, D_{IDr}). The full private key is known only by user and the full public key is also saved by KGC.
11
F、Offline
1) User ID_i selects $r_i \in Z_q{}^*$ randomly, and computes $R_i = r_iP$
G、Signcrypt
12
1) Computes $L_i = r_iP_{pub}Q_r$.
2) Computes $h_{ic} = h_1(ID_r, R_i, X_{IDr}, Y_{IDr}, L_i, \triangle)$.
3) Calculates $K_i = h_{ic} \oplus (m_i \,||\, ID_i)$.
4) Calculates $h_{ie} = h_2(ID_i, ID_r, R_i, K_i, X_{IDi}, Y_{IDi}, X_{ID}r, Y_{IDr})$.
5) Calculates $h = h_3(\triangle)$.
6) Calculates $W_i = DID_i + r_ih_{ie} + x_{IDi}h$.
7) Outputs ciphertext $C_i = (R_i, K_i, W_i)$.
Send ciphertext C_i
13
H、Aggregate
1)Computes $V = h_4(X_{IDr}w_1||X_{IDr}w_2||...||X_{IDr}w_n)$.
2)Computes $W = \sum_{i=1}^{n} w_i$.
3)Outputs the aggregate ciphertext $C = (V, W, R_1,..., R_n, K_1,..., K_n)$.
I、Verify
14
1)Computes $h_{ie} = h_2(ID_i, ID_r, R_i, K_i, X_{IDi}, Y_{IDi}, X_{IDr}, Y_{IDr})$ where $i = 1,..., n$.
2)Calculates $h = h_3(\triangle)$.
3)Calculates anti-collusion factor $b_i = P_{pub}Q_i + R_ih_{ic} + X_{IDi}h$, $V' = h_4(x_{IDr}b_1||x_{IDr}b_2||...||x_{IDr}b_n)$
4）ID_r determines collusion attack by checking if $V = V'$ is true.If not there's a collusive attack here.
5）Checks $WP = \sum_{i=1}^{n}(P_{pub}Q_i + R_ih_{ie} + X_{IDi}h)$, if the equation holds, it outputs true otherwise false.
15
J、Unsigncrypt
1) Calculates $L_i{}' = R_iD_{IDr}$.
2) Calculates $h_{ic}{}' = h_1(ID_r, R_i, X_{IDr}, Y_{IDr}, L_i{}', \triangle)$.
3) Calculates $ID_i \,||\, m_i{}' = K_i \oplus h_{ic}{}'$.
4) Outputs $\{m'_i\}_{i=1}^{n}$

Figure 4. Scheme process diagram.

Table 1. Description of symbols.

Symbol	Descritption
K	Security parameter
G	q–order additive group
P	Generator of G
m_{sk}	System master private key
P_{pub}	System master public key
V_i	Vehicle
R_{IDi}	Authentic identity of V_i
R_{IDr}	Authentic identity of receiver
P_{IDi}	Pseudonym of V_i
P_{IDr}	Pseudonym of receiver
(X_{IDi}, Y_{IDi})	Full public key of V_i
(x_{IDi}, D_{IDi})	Full private key of V_i
(X_{IDr}, Y_{IDr})	Full public key of receiver
(x_{IDr}, D_{IDr})	Full private key of receiver
n	Number of signcrypted messages
T_i	Timestamp

4.1. Set Up

Taking the security parameter K as input, the KGC and the TA are performed respectively as follows to generate system parameters.

- A q-order additive group G on the elliptic curve is chosen, where q is a 160-bit prime number, and its generator is a 512-bit prime number P;
- KGC selects m_{sk} stochastically from Zq^* as a master private key, and then the master public key is computed as $P_{pub} = m_{sk}P$;
- The TA selects $n \in Zq^*$ randomly, and computes $T_{pub} = nP$;
- Six cryptographic hash functions h_0 are chosen: $\{0,1\}^* \to Z_q^*$, h_1: $\{0,1\}^* \to \{0,1\}^n$ (n represents the bit-length of plaintexts), h_2: $\{0,1\}^* \to Z_q^*$, h_3: $\{0,1\}^* \to Z_q^*$, h_4: $G \to Z_q^*$, and h_5: $G \times \{0,1\}^* \to Z_q^*$;
- The KGC and the TA each secretly store m_{sk} and n and publish $Params$.
  $Params = \left\{q, P, P_{pub}, T_{pub}, h_0, h_1, h_2, h_3, h_4, h_5\right\}$.

4.2. Generation of Vehicle Pseudonym

This is performed jointly by the vehicle (V_i) with true identity RID_i and the TA (Algorithm 1).

- V_i selects $t_i \in Z_q^*$ randomly, and computes $PID_{i,1}$ = t_iP, O_i = $t_iT_{pub} \oplus RID_i$;
- V_i sends $PID_{i,1}$ and O_i to the TA, and the TA verifies the identity of V_i by checking if RID_i = $O_i \oplus nPID_{i,1}$ is true;
- The TA calculates another part of the pseudonym of V_i: $PID_{i,2} = RID_i \oplus h_5(nPID_{i,1}, T_i)$, and then sends the pseudonym $PID_i = \{PID_{i,1}, PID_{i,2}\}$ and the pseudonym's effective time T_i secretly to KGC.

Algorithm 1: Generation of vehicle pseudonym by the TA

```
Input: $t_i$
Output: $PID_i$
1. Computes $PID_{i,1} = t_i P$, $O_i = t_i T_{pub} \oplus RID_i$
2. Calculates $PID_{i,2} = RID_i \oplus h_5(nPID_{i,1}, T_i)$
3. Sends $PID_i = \{PID_{i,1}, PID_{i,2}\}$ to KGC
```

When a vehicle accedes to the network, it has to register to the TA with its own unique and authentic identity RID_i, and after that, the TA saves the $Params$ to the vehicle's OBU. Following the same method, the recipient's pseudonym can be obtained.

4.3. The User Key Generation

The user ID_i, which refers to the pseudonym of the vehicle PID_i, selects $x_{ID_i} \in Z_q^*$ stochastically as his or her secret value, then his or her public key is computed as $X_{ID_i} = x_{ID_i} P$.

4.4. Partial Key Extraction

- KGC selects $y_{ID_i} \in Z_q^*$ stochastically, and computes $Y_{ID_i} = y_{ID_i} P$ as his or her partial public key;
- The partial private key is computed by KGC $D_{ID_i} = m_{sk} Q_i$, where $Q_i = h_0(ID_i)$, and returns $\left(Y_{ID_i}, D_{ID_i}\right)$ to the user through the secure channel.

4.5. Private Key Verification

- By checking if $D_{ID_i} P = P_{pub} h_0(ID_i)$ is true, the user can verify the validity of its partial private key;
- Lastly, the whole public key of vehicle user ID_i is $\left(X_{ID_i}, Y_{ID_i}\right)$ and the whole private key is (x_{ID_i}, D_{ID_i}).

4.6. Offline

The user ID_i selects $r_i \in Z_q^*$ randomly, and R_i can be acquired as $R_i = r_i P$.

4.7. Signcrypt

This is executed by ID_i, which transmits message m_i to the recipient with the identity ID_r. By performing this algorithm, m_i can be signcrypted ($\triangle$ represents the trouble information of road conditions).

- The user figures out $L_i = r_i P_{pub} Q_r$;
- The user figures out $h_{ic} = h_1(ID_r, R_i, X_{ID_r}, Y_{ID_r}, L_i, \triangle)$;
- The user figures out part of the ciphertext $K_i = h_{ic} \oplus (m_i \parallel ID_i)$;
- The user calculates another hash function $h_{ie} = h_2(ID_i, ID_r, R_i, K_i, X_{ID_i}, Y_{ID_i}, X_{ID_r}, Y_{ID_r})$;
- Then calculates $h = h_3(\triangle)$;
- The user calculates another part of the ciphertext $W_i = D_{ID_i} + r_i h_{ie} + x_{ID_i} h$;
- Finally, the user obtains ciphertext $C_i = (R_i, K_i, W_i)$.

4.8. Aggregate

This is run by the fog device RSU (Algorithm 2).

- The RSU computes $V = h_4(X_{ID_r} W_1 \parallel X_{ID_r} W_2 \parallel \cdots \parallel X_{ID_r} W_n)$, which is used to verify whether there is a collusive attack;
- The RSU computes $W = \sum_{i=1}^{n} W_i$ to aggregate;
- Lastly, the RSU acquires the aggregate ciphertext $C = (V, W, R_1, \cdots, R_n, K_1, \cdots, K_n)$.

Algorithm 2: Aggregation by fog device RSU

```
Input: $X_{IDr}, C_i = (R_i, K_i, W_i)$
Output: $C$
1. Computes $V = h_4(X_{IDr}W_1||X_{IDr}W_2|| \ldots X_{IDr}W_n)$
2. Computes $W = \sum_{i=1}^{n} W_i$
3. Acquires the aggregate ciphertext $C = (V, W, R_1, \ldots, R_n, K_1, \ldots, K_n)$
```

4.9. Verification

This is carried out by the recipient ID_r.

- To begin with, the receiver computes $h_{ie} = h_2(ID_i, ID_r, R_i, K_i, X_{ID_i}, Y_{ID_i}, X_{ID_r}, Y_{ID_r})$ where $i = 1, \cdots, n$;
- Secondly, it calculates $h = h_3(\triangle)$;
- Then, in the most crucial step, the receiver calculates the anti-collusion factor $b_i = P_{pub}Q_i + R_ih_{ie} + X_{ID_i}h$, $V' = h_4(x_{ID_r}b_1 \parallel x_{ID_r}b_2 \parallel \cdots \parallel x_{ID_r}b_n)$;
- The receiver ID_r determines the collusion attack by checking if $V = V'$ is true. If not, there is a collusive attack here;
- In the end, the receiver checks $WP = \sum_{i=1}^{n} \left(P_{pub}Q_i + R_ih_{ie} + X_{ID_i}h \right)$, and if the equation holds, validation is successful.

4.10. Unsigncryption

This is carried out by the recipient ID_r (if all the above verifications are passed) (Algorithm 3).

- First, the receiver computes $L_i' = R_iD_{ID_r}$;
- Second, using the given parameters, it calculates $h_{ic}' = h_1(ID_r, R_i, X_{ID_r}, Y_{ID_r}, L_i', \triangle)$;
- Then, the receiver can obtain $m_i' \parallel ID_i = K_i \oplus h_{ic}'$;
- Finally, it obtains $\{m_i' \parallel ID_i\}_{i=1}^{n}$.

Algorithm 3: Unsigncryption by recipient ID_r

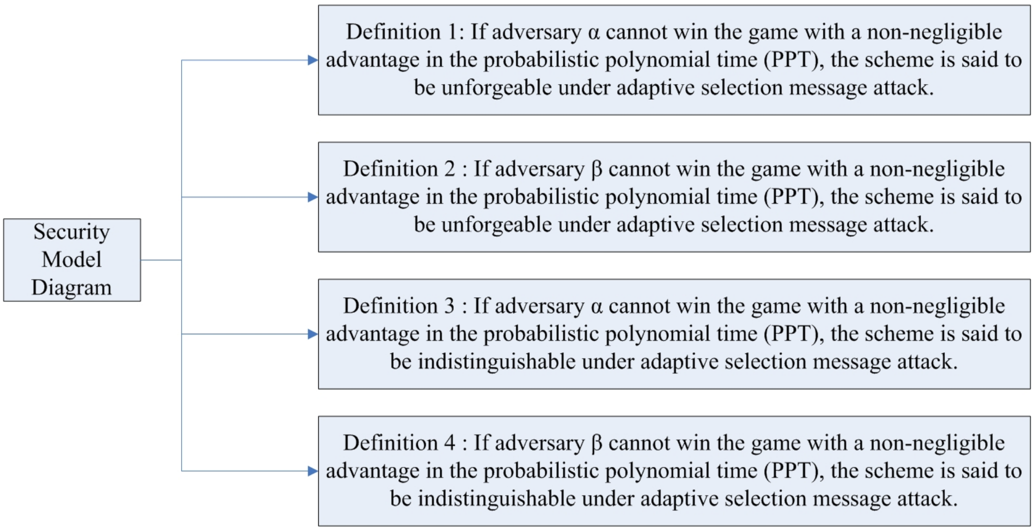
```
Input: $C$
Output: $\{m_i'||ID_i\}_{i=1}^{n}$
1. Computes $L_i' = R_iD_{ID_r}$
2. Computes $h_{ic}' = h_1(ID_r, R_i, X_{ID_r}, Y_{ID_r}, L_i', \triangle)$
3. Obtains $\{m_i'||ID_i\}_{i=1}^{n}$
```

5. Security Analysis

In this part, our scheme is proven to be safe under the security model and then it is shown to achieve our design goals.

5.1. Security Model

From the scheme in [10], we can find two kinds of adversaries: α and β, which are external and inner attackers. Meanwhile, our security model diagram is shown in Figure 5.



Definition 1: If adversary α cannot win the game with a non-negligible advantage in the probabilistic polynomial time (PPT), the scheme is said to be unforgeable under adaptive selection message attack.

Definition 2 : If adversary β cannot win the game with a non-negligible advantage in the probabilistic polynomial time (PPT), the scheme is said to be unforgeable under adaptive selection message attack.

Definition 3 : If adversary α cannot win the game with a non-negligible advantage in the probabilistic polynomial time (PPT), the scheme is said to be indistinguishable under adaptive selection message attack.

Definition 4 : If adversary β cannot win the game with a non-negligible advantage in the probabilistic polynomial time (PPT), the scheme is said to be indistinguishable under adaptive selection message attack.

Figure 5. Security model diagram.

5.2. Provable Security

- Correctness: First, we prove the proposed scheme to be correct.
  a. User authentication
  $RID_i = O_i \oplus nPID_{i,1}$
  $RID_i = t_i T_{pub} \oplus RID_i \oplus nt_i P$
  $RID_i = t_i T_{pub} \oplus RID_i \oplus t_i T_{pub}$
  $RID_i = RID_i;$
  b. Private key verification
  $D_{IDi}P = P_{pub}h_0(ID_i)$
  $D_{IDi}P = m_{sk}Ph_0(ID_i)$
  $D_{IDi}P = D_{IDi}P;$
  c. Verification of the existence of anti collusion attacks
  $V = h_4(X_{IDr}W_1 \parallel X_{IDr}W_2 \parallel \cdots \parallel X_{IDr}W_n)$
  $V = h_4(x_{IDr}PW_1 \parallel x_{IDr}PW_2 \parallel \cdots \parallel x_{IDr}PW_n)$
  $W_1 = D_{ID1} + r_1 h_{ie} + x_{ID1}h, PW_1 = b_1$
  $W_2 = D_{ID2} + r_2 h_{ie} + x_{ID2}h, PW_2 = b_2$
  $W_n = D_{IDn} + r_n h_{ie} + x_{IDn}h, PW_n = b_n;$
  $V = h_4(x_{IDr}b_1 \parallel x_{IDr}b_2 \parallel \cdots \parallel x_{IDr}b_n) = V'$
  d. Unsigncrypt verification
  $m_i' \parallel ID_i = k_i \oplus h_{ic}'$
  $m_i' \parallel ID_i = h_{ic} \oplus (m_i \parallel ID_i) \oplus h_{ic}'$
  $m_i' \parallel ID_i = m_i \parallel ID_i.$
- Resisting collusion attacks: Second, we demonstrate that our scheme can resist collusion attacks.

Proof. Challenger $\wp$'s goal is to use adversary γ to break the collision resistance of hash function h_4. □

Setup: Challenger $\wp$ imports safety parameter l, sets a as the master private key, calculates $P_{pub} = aP$, runs the setup algorithm to form system parameters $prms = (q, P, G, P_{pub}, T_{pub}, h_0, h_1, h_2, h_3)$, and sends them to γ.

Query phrase: γ adaptively executes the following oracle machine queries. For better performances, $\wp$ maintains the following six lists: L_{h0}, L_{h1}, L_{h2}, L_{h3}, L_s, L_{sk}, and L_{pk}, L_{rp}, which are used to record the query data of γ for h_0, h_1, h_2, and h_3, as well as secret value extraction, partial private key extraction, public key extraction, and public key replacement. Initially they are empty.

h_0 queries: When γ executes h_0 queries for ID_i, Q_i turns back to γ if there is a relational tuple (ID_i, Q_i, c_i) in the list L_{h0}; if not, $\wp$ stochastically chooses $c_i \in \{0,1\}$, and sets $P_r[c_i = 1] = \delta$, $P_r[c_i = 0] = 1 - \delta$. If $c_i = 0$, $\wp$ stochastically selects $Q_i \in Z_q^*$, and adds $(ID_i, Q_i, 0)$ to the list L_{h0}, and then turns Q_i back; if $c_i = 1$, $\wp$ creates $Q_i = k$, adds $(ID_i, k, 1)$ to the list L_{h0}, and turns Q_i back.

h_1 queries: When γ executes h_1 queries for $(ID_r, R_i, X_{ID_r}, Y_{ID_r}, L_i, \triangle)$, h_{ic} is turned back to γ if there is a relational tuple $(ID_r, R_i, X_{ID_r}, Y_{ID_r}, L_i, \triangle, h_{ic})$ in the list L_{h1}; if not, $\wp$ stochastically selects $h_{ic} \in Z_q^*$, adds $(ID_r, R_i, X_{ID_r}, Y_{ID_r}, L_i, \triangle, h_{ic})$ to the list L_{h1}, and turns h_{ic} back.

h_2 queries: When γ executes h_2 queries for $(ID_i, ID_r, R_i, K_i, X_{ID_i}, Y_{ID_i}, X_{ID_r}, Y_{ID_r})$, then h_{ie} is turned back to γ. If there is no relational tuple $(ID_i, ID_r, R_i, K_i, X_{ID_i}, Y_{ID_i}, X_{ID_r}, Y_{ID_r}, h_{ie})$ in the list L_{h2}, $\wp$ stochastically selects $h_{ie} \in Z_q^*$, adds $(ID_i, ID_r, R_i, K_i, X_{ID_i}, Y_{ID_i}, X_{ID_r}, Y_{ID_r}, h_{ie})$ to list the L_{h2}, and turns h_{ie} back.

h_3 queries: When γ executes h_3 queries for $\triangle$, h is turned back to γ if there is a relational tuple $(\triangle, h)$ in the list L_{h3}; if not, $\wp$ stochastically selects $h \in Z_q^*$, adds $(\triangle, h)$ to the list L_{h3}, and turns h back.

ExtractSecretValue queries: When γ executes ExtractSecretValue queries for ID_i, x_{ID_i} is turned back to γ if there is a relational tuple (ID_i, x_{ID_i}) in the list L_s; if not, $\wp$ stochastically selects $x_{ID_i} \in Z_q^*$, adds (ID_i, x_{ID_i}) to the list L_s, and turns x_{ID_i} back.

ExtractPartialPrivateKey queries: When γ performs ExtractPartialPrivateKey queries for (ID_i, X_{ID_i}), (Y_{ID_i}, D_{ID_i}) is turned back to γ if there is a relational tuple $(ID_i, X_{ID_i}, Y_{ID_i}, D_{ID_i})$ in the list L_{sk}; if not, $\wp$ stochastically selects D_{ID_i}, $Y_{ID_i} \in Z_q^*$, adds $(ID_i, X_{ID_i}, Y_{ID_i}, D_{ID_i})$ to the list L_{sk}, and turns (Y_{ID_i}, D_{ID_i}) back.

ExtractPublicKey queries: When γ executes ExtractPublicKey queries for ID_i, (X_{ID_i}, Y_{ID_i}) is turned back to γ if there is a relational tuple $(ID_i, X_{ID_i}, Y_{ID_i})$ in the list L_{pk}; if not, $\wp$ stochastically selects $x_{ID_i} \in Z_q^*$, calculates $X_{ID_i} = x_{ID_i}P$, and then executes ExtractPartialPrivateKey queries to acquire $(ID_i, X_{ID_i}, Y_{ID_i}, D_{ID_i})$, adds $(ID_i, X_{ID_i}, Y_{ID_i})$ to the list L_{pk}, and turns (X_{ID_i}, Y_{ID_i}) back.

ReplacePublicKey queries: γ chooses a new public key (X'_{ID_i}, Y'_{ID_i}) to substitute for the initial public key (X_{ID_i}, Y_{ID_i}).

Signcrypt queries: When γ executes signcrypt queries for (ID_i, ID_r, m_i), $\wp$ queries (ID_i, Q_i, c_i) in the list L_{h0}. If $c_i = 0$, $\wp$ performs the Signcrypt algorithm, signcrypts m_i, and turns the ciphertext $C_i = (R_i, K_i, W_i)$ back to γ; if $c_i = 1$, $\wp$ stochastically chooses r'_i, $a_1 \in Z_q^*$, extracts x_{ID_i}, h_{ic}, Q_i, h_{ie}, and h from querying the lists L_s, L_{h0}, L_{h1}, L_{h2}, and L_{h3}. Then, $\wp$ calculates $R_i = r'_iP$, $W_i = a_1 + r'_ih_{ie} + x_{ID_i}h$, and $K_i = h_{ic} \oplus (m_i \parallel ID_i)$ and turns the ciphertext $C_i = (R_i, K_i, W_i)$ back to γ.

Forge Phrase: After the queries, it is easy for us to know that each of the users ID_u can generate $C^* = \left(V^*, W^*, R_1^*, R_2^*, \ldots, Rn^*, K_1^*, K_2^*, \ldots, K_n^*\right)$, and $V^* = h_4(X_{ID_u}W_1 \parallel X_{ID_u}W_2 \parallel \cdots \parallel X_{ID_u}W_n)$. If $V^* = V' = h_4(x_{ID_u}b_1 \parallel x_{ID_u}b_2 \parallel \cdots \parallel x_{ID_u}b_n)$, in which $b_i = P_{pub}Q_i + R_ih_{ie} + X_{ID_i}h$. Assuming C_i^* is an invalid signcryption, then $W_i^*P \neq P_{pub}Q_i^* + R_i^*h_{ie}^* + X_{ID_i}^*h$, so $X_{ID_u}W_i^* \neq x_{ID_u}\left[P_{pub}Q_i + R_ih_{ie} + X_{ID_i}h\right]$, therefore including the two different inputs $X_{ID_u}W_i^*$ and $x_{ID_u}[P_{pub}Q_i + R_ih_{ie} + X_{ID_i}h]$; however, $h_4(X_{ID_u}W_1||X_{ID_u}W_2 \parallel \ldots||X_{ID_u}W_n) = h_4(x_{ID_u}[P_{pub}Q_1 + R_1h_{ie} + X_{ID_1}h] \parallel x_{ID_u}[P_{pub}Q_2 + R_2h_{ie} + X_{ID_2}h] \parallel \ldots \parallel x_{ID_u}[P_{pub}Q_n + R_nh_{ie} + X_{ID_n}h])$. For this reason, $\wp$ successfully solved the collision resistance problem of h_4.

- Unforgeability and confidentiality: Additionally, the proofs that our scheme can realize unforgeability and confidentiality under attacks from these two kinds of adversaries are the same as for the scheme in [23] and in [10]; thus, the concrete procedures are not given here.

5.3. Analysis of Security Requirements

Lastly, we prove that our scheme satisfies the security demands detailed in Section 2.3.

- Mutual authentication: Participants can identify each other by checking if the message they receive is activating;
- Vehicle identity verification: The decrypted plaintext package contains the user's pseudonym. At this time, the TA can trace the user's true identity back to the user's real identity through the pseudonym, its own private key, and timestamp to complete user identity verification, and this operation can only be performed by the TA;
- User anonymity: Due to the use of pseudonyms throughout the entire process and the fact that the user's true identity can only be acquired by trusted institutions, the anonymity of the user's identity is ensured;
- Untraceability: The user chooses the stochastic number r_i, calculates $R_i = r_i P$ and $W_i = D_{ID_i} + r_i h_{ie} + x_{ID_i} h$, and then computes K_i to acquire ciphertext C_i. Hence, the attackers cannot follow the trail of vehicle users. At the same time, due to the use of pseudonyms throughout the entire process, attackers will not track information related to the user's identity;
- Nondeniability: The received message contains the identity ID_i of the sender and ID_r of the receiver, and the identity ID_i of the sender is also included in the decryption result, so the vehicle user cannot deny participation.
- Resist collusion attacks: Because of the collision resistance of the hash function, collusion attacks can be detected by utilizing the anti-collusion factor.

6. Performance Analysis

Performance analysis is introduced in three parts in this section. The description includes the functionality, computational cost, and communication cost. In VANETs, ensuring the confidentiality, validity, and vehicle identity of messages is crucial, and achieving fast authentication in resource-constrained environments is also essential. Due to the fact that the schemes in [19,23,28–30] basically have the above functions, it is more meaningful to compare the proposed schemes with them in the context of vehicle networking. However, based on the scheme in [23], we propose a new scheme that can resist collusion attacks, due to the fact that our scheme and the scheme in [23] have similar communication and computational costs, and we only compare our plan with the scheme in [19,28–30] here.

6.1. Functionality

Message Confidentiality, Message Verifiability, Vehicle Verifiability, Key Escrow Resilience,Quick Verification, and Resist Collusion Attack are all supported by our approach alone, as shown in Table 2 (× represents not having this feature, while the opposite is true for the √). The schemes in [28,29] cannot resist collusion attacks and do not provide vehicle verifiability. Although the scheme in [30] can support vehicle verifiability and resist collusion attacks, it cannot ensure the confidentiality of messages because it is an aggregate signature scheme. Thus, from here on the scheme in [30] will only be discussed about communication costs.

Table 2. Functional comparison.

Schemes	[28]	[23]	[29]	[30]	[19]	Our Scheme
Message Confidentiality	√	√	√	×	√	√
Message Verifiability	√	√	√	√	√	√
Vehicle Verifiability	×	×	×	√	√	√
Key Escrow Resilience	√	√	√	√	√	√
Quick Verification	√	√	√	√	√	√
Resist Collusion Attack	×	×	×	√	×	√

6.2. Computational Cost

The specific experimental environment for our simulation is shown in Table 3. We used an Apple M1, 8 GB, macOS Big Sur 11.6 operating system; the code running environment was Ubuntu 18. By using MIRACL 5.5.4 and JPBC 2.0.0, we obtained the time of one laborious operation.

One laborious operation is taken into consideration, as indicated in Table 4. The vehicle user in the scheme in [28] conducts seven scalar multiplications during the signcryption stage, whereas the unsigncrypt method performs two scalar multiplications. The overall costs in [28] comprise nine scalar multiplications. The vehicle user conducts three scalar multiplications in the signcryption stage of the scheme in [29], and the unsigncrypt algorithm likewise performs three scalar multiplications. The overall cost in [29] includes six scalar multiplications. In scheme [19], the user conducts three scalar multiplications in the signcryption stage, four scalar multiplications in the unsigncryption stage, and the total costs are more than six scalar multiplications. The vehicle user only performs one scalar multiplication throughout the signcryption procedure for our scheme, the RSU must perform one scalar multiplication, and the overall cost incurred only consists of two scalar multiplications.

The calculation costs at different stages are compared in Figure 6, Table 5, and the entire computation costs are compared in Figure 7, Table 5. We selected $n = 5000$, which represents the number of signcrypted messages. It is evident from the figure that our scheme has the lower computational overhead in every stage.

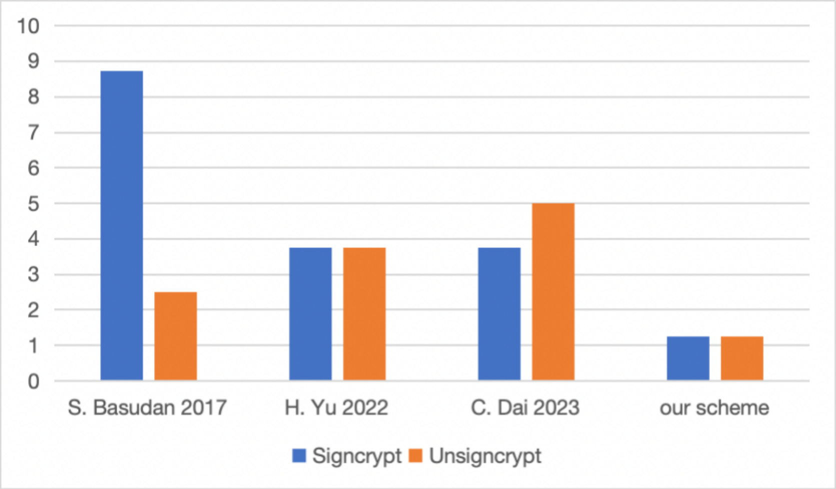


Figure 6. Comparison of computation costs in different stages [19,28,29].

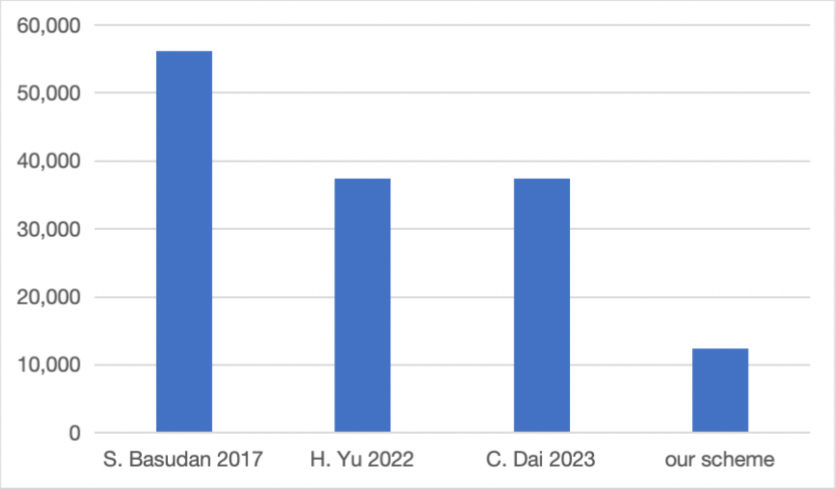


Figure 7. Comparison of total computation costs [19,28,29].

Table 3. Experimental environment.

Hardware	CPU	Apple M1
Hardware	Memory	8 GB
Software	Operating System	macOS Big Sur 11.6
Software	Program Language	C 17 and JAVA 1.8.0
Software	Library	MIRACL 5.5.4 and JPBC 2.0.0

Table 4. Operation time.

Symbol	Computing Operation	Executing Time (ms)
t_s	Scalar multiplication in G	1.248

Table 5. Computational cost Analysis.

Schemes	Signcrypt	Unsigncrypt	Total Cost
[28]	$7t_s$	$2t_s$	$9nt_s$
[29]	$3t_s$	$3t_s$	$6nt_s$
[19]	$3t_s$	$4t_s$	$(6n+1)t_s$
Our Scheme	$1t_s$	$1t_s$	$2nt_s$

6.3. Communication Cost

Assume that $|G| = 160$ bit, $n = 5000$, and $|m| = 512$ bit respectively indicate the length of values of G and the message sent. The scheme in [30] can resist collusion attacks but cannot ensure the confidentiality of the message. At the same time, sending the message and aggregated signature results together to the receiver incurs significant communication costs. The receiver must obtain the ciphertext $C = (R_1, \ldots, R_n, K_1, \ldots, K_n, W)$ in the schemes in [28,29]. In the scheme in [19], the communication costs are the lowest. While our scheme undoubtedly increases the size of the signcryption result due to the verification factors for resisting collusion attacks, it also fully complies with the requirements of VANETs' features and is even more secure than the schemes in [19,28,29]. Nevertheless, in order to verify if there is a collusion attack, the receiver must receive the ciphertext $C = (R_1, \ldots, R_n, K_1, \ldots, K_n, W, V)$. To sum up, the suggested plan is suitable for VANETs. The details are shown in Table 6. In order to further evaluate the proposed scheme, the relationship between the vehicle density and communication cost under different schemes is shown in Figure 8. It can be seen that as the density of vehicles increases, the communication overhead also increases, which is in line with the scenario of vehicle networking. The scheme proposed in this paper has certain advantages over the scheme in [30] and is closer to other schemes. Without adding too much overhead, it also fully complies with the requirements of VANET's features and is even more secure.

Table 6. Communication cost analysis.

Schemes	Communication Cost
[19]	$n\|G\| + n\|m\|$
[28]	$(n+1)\|G\| + n\|m\|$
[29]	$(n+1)\|G\| + n\|m\|$
[30]	$(n+7)\|G\| + n\|m\|$
Our Scheme	$(n+2)\|G\| + n\|m\|$

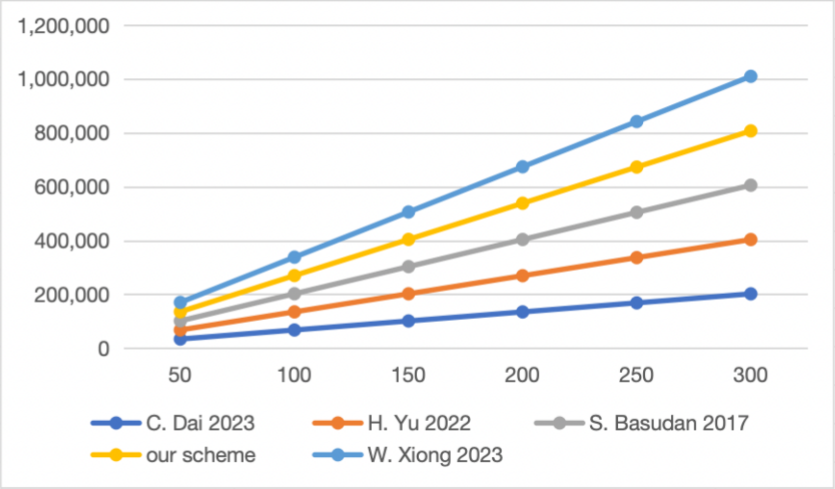


Figure 8. Comparison of communication costs for different vehicle densities [19,28–30].

7. Conclusions

In order to provide a safer environment for the Internet of Vehicles, we propose a brand-new certificateless online/offline aggregate signcryption scheme against collusion attacks based on fog computing. It can successfully enable mobility, location awareness, and low latency. Mutual authentication, anonymity, verifiability, untraceability, and secrecy are only a few of the security requirements that the scheme can satisfy. Due to the significance of security in VANETs, we establish the security of the scheme in the random oracle model. The security analysis and performance study demonstrate that our scheme can resist collusion attacks without raising computational complexity, which is crucial for resource-constrained VANETs. Concurrently, the data provided by our scheme, which meet the requirements of confidentiality and real-time transmission, are essential to achieving reliable driving in all types of weather and road conditions. Therefore, our method is suitable for the Internet of Vehicles and completely aligns with VANET criteria. However, the communication expenses of our scheme are a little high. Future research will work even harder to cut communication costs, such as by shortening the ciphertext length without affecting the effect. At the same time, we will focus on how to effectively recognize coordinated assaults in order to further enhance the security of the internet of vehicles.

Author Contributions: Methodology, W.Z. and S.L.; Writing—review & editing, Y.L., J.C., B.F. and Y.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

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

References

1. Ros, F.J.; Martinez, J.A.; Ruiz, P.M. A survey on modeling and simulation of vehicular networks: Communications mobility and tools. *Comput. Commun.* **2014**, *43*, 1–15. [CrossRef]
2. Satyanarayanan, M.; Bahl, P.; Caceres, R.; Davies, N. The Case for VM-based Cloudlets in Mobile Computing. *IEEE Pervasive Comput.* **2011**, *8*, 14–23. [CrossRef]
3. Bessis, N.; Dobre, C. *Big Data and Internet of Things: A Roadmap for Smart Environments*; Springer: Cham, Switzerland, 2014.
4. Hu, P.; Ning, H.; Qiu, T.; Song, H.; Wang, Y.; Yao, X. Security and privacy preservation scheme of face identification and resolution framework using fog computing in Internet of Things. *IEEE Internet Things J.* **2017**, *4*, 1143–1155. [CrossRef]
5. 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]
6. Li, L.; Li, Y.; Hou, R. A novel mobile edge computing-based architecture for future cellular vehicular networks. In Proceedings of the 2017 IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, USA, 19–22 March 2017; pp. 1–6.

7. Wang, G.; Wang, J.; Guo, Z. Online/offline self-updating encryption. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2016**, *99*, 2517–2526. [CrossRef]
8. Yuan, Y. Security Analysis of an Enhanced Certificateless Signcryption in the Standard Model. *Wirel. Pers. Commun.* **2020**, *112*, 1. [CrossRef]
9. Al-Riyami, S.; Paterson, K. Certificateless public key cryptography. *Proc. Asiacrypt.* **2003**, *2894*, 452–473.
10. Eslami, Z.; Pakniat, N. Certificateless aggregate signcryption: Security model and a concrete construction secure in the random oracle model. *J. Comput. Inf. Sci.* **2014**, *26*, 276–286. [CrossRef]
11. Zhang, L.; Hu, C.; Wu, Q.; Domingo-Ferrer, J.; Qin, B. Privacy-Preserving Vehicular Communication Authentication with Hierarchical Aggregation and Fast Response. *IEEE Trans. Comput.* **2016**, *65*, 2562–2574. [CrossRef]
12. Sun, D.; Mu, Y.; Susilo, W. A Generic Construction of Identity-Based Online/Offline Signcryption. In Proceedings of the 2008 IEEE International Symposium on Parallel and Distributed Processing with Applications, Sydney, NSW, Australia, 10–12 December 2008; pp. 707–712.
13. Sabireen, H.; Neelanarayanan, V. A Review on Fog Computing: Architecture, Fog with IoT, Algorithms and Research Challenges. *ICT Express* **2021**, *7*, 2. [CrossRef]
14. Dastjerdi, A.V.; Gupta, H.; Calheiros, R.N.; Ghosh, S.K.; Buyya, R. *Fog Computing: Principles Architectures and Applications*; Morgan Kaufmann: Burlington, MA, USA, 2016; pp. 61–75.
15. Erskine, S.K.; Elleithy, K.M. Secure Intelligent Vehicular Network Using Fog Computing. *Electronics* **2019**, *8*, 4. [CrossRef]
16. Nayak, P.; Swapna, G. Security issues in IoT applications using certificateless aggregate signcryption schemes: An overview. *IEEE Internet Things J.* **2023**, *21*, 100641. [CrossRef]
17. Lu, H.; Xie, Q. An efficient certificateless aggregate signcryption scheme from pairings. In Proceedings of the 2011 International Conference on Electronics, Communications and Control (ICECC), Ningbo, China, 9 September 2011; pp. 132–135.
18. Kim, T.H.; Kumar, G.; Saha, R.; Alazab, M.; Buchanan, W.J.; Rai, M.K.; Geetha, G.; Thomas, R. CASCF: Certificateless Aggregated SignCryption Framework for Internet-of-Things Infrastructure. *IEEE Access* **2020**, *8*, 94748–94756. [CrossRef]
19. Dai, C.; Xu, Z. Pairing-Free Certificateless Aggregate Signcryption Scheme for Vehicular Sensor Networks. *IEEE Internet Things J.* **2023**, *10*, 5063–5072. [CrossRef]
20. Hou, Y.; Cao, Y.; Xiong, H.; Song, Y.; Xu, L. An Efficient Online/Offline Heterogeneous Signcryption Scheme with Equality Test for IoVs. *IEEE Trans. Veh. Technol.* **2023**, *72*, 12047–12062. [CrossRef]
21. Lai, J.; Mu, Y.; Guo, F. Efficient identity-based online/offline encryption and signcryption with short ciphertext. *Int. J. Inf. Security* **2017**, *16*, 299–311. [CrossRef]
22. An, H.; He, D.; Peng, C.; Luo, M.; Wang, L. Efficient Certificateless Online/Offline Signcryption Scheme without Bilinear Pairing for Smart Home Consumer Electronics. *IEEE Trans. Consum. Electron.* **2023**, 1. [CrossRef]
23. 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]
24. Pan, S.; Wang, S. Certificateless aggregate signcryption scheme against forgery attacks for vehicular ad hoc networks. *J. Xidian Univ.* **2023**, *50*, 169–177.
25. Alice, G.S.; Maria, S.S.; Arunita, J. Blockchain-Based Pseudonym Management Scheme for Vehicular Communication. *Electronics* **2021**, *10*, 13.
26. Wang, D.; Cheng, H.; Wang, P.; Huang, X.; Jian, G. Zipf's law in passwords. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2776–2791. [CrossRef]
27. Shabani, F.; Gharaee, H.; Ghaffari, F. An intelligent RFIDenabled authentication protocol in VANET. In Proceedings of the 2018 9th International Symposium on Telecommunications (IST), Tehran, Iran, 17–19 December 2018; pp. 587–591.
28. Basudan, S.; Lin, X.; Sankaranarayanan, K. A privacy-preserving vehicular crowdsensing-based road surface condition monitoring system using fog computing. *IEEE Internet Things J.* **2017**, *4*, 772–782. [CrossRef]
29. Yu, H.; Ren, R. Certificateless Elliptic Curve Aggregate Signcryption Scheme. *IEEE Syst. J.* **2022**, *16*, 2347–2354. [CrossRef]
30. Xiong, W.; Wang, R.; Wang, Y.; Wei, Y.; Zhou, F.; Luo, X. Improved Certificateless Aggregate Signature Scheme against Collusion Attacks for VANETs. *IEEE Syst. J.* **2023**, *17*, 1098–1109. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.