entropy

MDPI

Article

A Conditional Privacy-Preserving Identity-Authentication Scheme for Federated Learning in the Internet of Vehicles

Shengwei Xu [1,*] and Runsheng Liu [2]

[1] Institute of Information Security, Beijing Electronic Science and Technology Institute, Beijing 100070, China
[2] Department of Cryptography Science and Technology, Beijing Electronic Science and Technology Institute, Beijing 100070, China; besti_paradise@163.com
* Correspondence: 18510529691@163.com

Abstract: With the rapid development of artificial intelligence and Internet of Things (IoT) technologies, automotive companies are integrating federated learning into connected vehicles to provide users with smarter services. Federated learning enables vehicles to collaboratively train a global model without sharing sensitive local data, thereby mitigating privacy risks. However, the dynamic and open nature of the Internet of Vehicles (IoV) makes it vulnerable to potential attacks, where attackers may intercept or tamper with transmitted local model parameters, compromising their integrity and exposing user privacy. Although existing solutions like differential privacy and encryption can address these issues, they may reduce data usability or increase computational complexity. To tackle these challenges, we propose a conditional privacy-preserving identity-authentication scheme, CPPA-SM2, to provide privacy protection for federated learning. Unlike existing methods, CPPA-SM2 allows vehicles to participate in training anonymously, thereby achieving efficient privacy protection. Performance evaluations and experimental results demonstrate that, compared to state-of-the-art schemes, CPPA-SM2 significantly reduces the overhead of signing, verification and communication while achieving more security features.

Keywords: federated learning; Internet of Vehicles; authentication; certificateless-based cryptography

check for updates

Citation: Xu, S.; Liu, R. A Conditional Privacy-Preserving Identity-Authentication Scheme for Federated Learning in the Internet of Vehicles. *Entropy* **2024**, *26*, 590. https://doi.org/10.3390/e26070590

Academic Editor: Boris Ryabko

Received: 4 June 2024
Revised: 27 June 2024
Accepted: 4 July 2024
Published: 10 July 2024

1. Introduction

With the rapid development of intelligent transportation systems and Internet of Things (IoT) technology, the Internet of Vehicles (IoV) has become an essential component of smart cities [1]. IoV enables real-time sharing of traffic information and intelligent coordination of vehicles through communication between vehicles and between vehicles and infrastructure. Additionally, with the advancement of machine learning technology, many automotive companies are leveraging machine learning in the IoV to provide more intelligent and efficient services to users [2]. By collecting a large amount of vehicle data to train models, they offer applications such as autonomous driving and traffic flow prediction [3]. However, traditional centralized model training requires gathering vehicle data to the central server for training. Since this vehicle data often contains a significant amount of personal information, such as driving habits, travel routes, home and work locations, many users are concerned about privacy breaches and are reluctant to send their data to the central server [4]. Moreover, recent data security regulations prohibit automotive companies from collecting user data without authorization. To address these privacy concerns, federated learning (FL) has emerged as a solution [5]. FL is a decentralized machine learning approach where multiple clients (such as smartphones, vehicles or other devices) collaboratively train a shared model under the orchestration of a central server while keeping the data localized [6]. Instead of sending raw data to a central server, each client processes the data locally and only shares the model updates (like gradients or parameters) with the central server. The server then aggregates these updates to form a global model. Currently, FL has been widely applied in various IoV scenarios, such as trajectory prediction, advanced driver-assistance systems and traffic flow prediction and management [7].

Although FL addresses the issue of data silos, researchers have found that without proper protection of the transmitted model parameters, attackers can still infer privacy information about user data [8]. Additionally, during the aggregation of parameters by the central server, there is a risk that the server may attempt to infer original data information from the uploaded model parameters. Moreover, due to the open nature of the IoV, attackers can easily eavesdrop on and manipulate messages transmitted between vehicles, gaining access to the vehicles' real identities and further tracking their behaviors, posing a threat to user privacy [9].

To address the issue of privacy leakage in federated learning, existing solutions are mainly categorized into differential privacy (DP) [10–12] and encryption techniques [13–18]. DP protects the privacy of original data by adding random noise to model parameters. Wei et al. [10] proposed a differential privacy-based federated learning framework, which achieves different levels of differential privacy protection by adding artificial noise to client parameters before aggregation. Zhao et al. [11] combined DP with federated learning, proposing four localized differential privacy mechanisms to perturb gradients generated by vehicles, thereby preventing privacy leakage. Zhou et al. [12] achieved high-level privacy protection by adding noise and theoretically proved the convergence of their algorithm. Although DP-based solutions have been extended to all machine learning algorithms in deep learning, the added random noise can degrade model accuracy and extend the model convergence time. Encryption-based solutions can be divided into homomorphic encryption and secure multiparty computation (SMC). Zhou et al. [13] combined differential privacy, blinding and Paillier homomorphic encryption to resist model attacks and achieve secure aggregation of model parameters. Ma et al. [14] proposed a dual-trapdoor homomorphic encryption scheme, ShieldFL, which can defend against model poisoning attacks and protect privacy. They also introduced a secure cosine similarity method for Byzantine-robust aggregation. Hijazi et al. [15] introduce four different fully homomorphic encryption (FHE)-based methods for FL, which securely transmit model parameters in encrypted form, thereby enhancing robust privacy and security protection. Zhang et al. [16] present a lightweight dual-server secure aggregation protocol based on secret sharing, achieving both privacy protection and Byzantine robustness. A typical example is secret sharing. This method reduces computational overhead compared to homomorphic encryption but increases the number of communication rounds and communication overhead, thereby hindering the training efficiency of federated learning. Furthermore, encryption-based solutions prevent the cloud server from directly accessing plaintext local model parameters during aggregation. This hinders integration with Byzantine-robust federated learning defense mechanisms [17,18], as existing Byzantine-robust defense mechanisms focus on computing similarities directly on plaintext model parameters. Therefore, it is necessary to research a privacy-preserving federated learning solution suitable for the IoV that can balance efficiency and practicality.

To ensure the authenticity and integrity of communication data in the IoV, many identity-authentication protocols have been proposed [19]. Currently, existing identity-authentication protocols in the IoV can be primarily categorized into three types: public key infrastructure-based (PKI-based) [20], identity-based (ID-based) [21–24] and certificateless-based [25–28]. PKI-based identity-authentication protocols bind a vehicle's identity to its public key through digital certificates. Vehicles use their private keys to sign messages, and verifiers use the public keys from the vehicle's digital certificates to verify the signatures. The main drawback of this method is the significant storage and maintenance overhead associated with managing a large number of digital certificates and certificate revocation lists. Identity-based authentication protocols directly use the vehicle's identity information as the public key, thereby avoiding the overhead of certificate management and maintenance. Zhao et al. [22] proposed an identity-based federated learning collaborative authentication protocol for shared data, achieving efficient anonymous authentication and key agreement between vehicles and other entities. Zhang et al. [23] proposed an ID-based conditional privacy-preserving identity-authentication scheme that does not require bilinear pairings or hash-to-point operations, enabling efficient vehicle authentication. Kanchan et al. [24] proposed a federated learning algorithm based on group signatures, enhancing the protection of node identities. Although ID-based identity-authentication schemes can achieve efficient vehicle authentication, they have the issue of key escrow. Therefore, certificateless identity-authentication schemes have been proposed as a promising solution. However, this approach has a key escrow problem, as the Trusted Authority (TA) has full control over the vehicle's private keys and can generate legitimate signatures for any vehicle. To address the key escrow issue, certificateless authentication protocols have been proposed. In these protocols, a vehicle's private key consists of two parts: one part is a secret value selected by the vehicle itself, and the other part is a partial private key generated by TA. Lin et al. [25] proposed a certificateless authentication and key agreement protocol for IoV based on blockchain. This protocol utilizes the decentralized architecture of blockchain to achieve decentralized trusted third-party services, thus mitigating issues such as single-point failure and the risk of trusted third-party disclosure. It aims to achieve efficient authentication between vehicles. Jiang et al. [26] proposed a certificateless anonymous identity-authentication scheme, which aims to anonymize the relationship between terminal identities and data. However, the use of bilinear pairing operations affects authentication efficiency. Ma et al. [27] extended Jiang's work by proposing a certificateless identity-authentication scheme that does not require bilinear pairing operations and supports batch verification. However, this scheme lacks dynamic member-management capabilities, and the pseudonyms generated by vehicles cannot be dynamically updated. Currently, most existing certificateless authentication protocols use bilinear pairing operations or do not support batch verification, leading to low authentication efficiency. Additionally, most certificateless authentication protocols are independently designed and are not integrated with existing international standard cryptographic algorithms, making them inconvenient for practical application and widespread adoption. Therefore, it is necessary to study an efficient authentication protocol to establish a secure communication environment for the IoV.

To address the aforementioned challenges, we propose a conditional privacy-preserving authentication scheme called CPPA-SM2, which provides secure authentication and privacy protection for vehicle communication and federated learning in the IoV. Specifically, it is based on the fact that if vehicles send messages and participate in training anonymously, even if attackers or the cloud server obtain the plaintext local model parameters and infer some data information, they cannot associate this information with a specific real vehicle identity, thus achieving privacy protection. Our main contributions are as follows:

- We propose a Conditional Privacy-Preserving Authentication scheme, CPPA-SM2, and integrate it with federated learning. Vehicles participate in federated learning training anonymously, obfuscating the link between local model parameters and the vehicle's real identity, thus achieving privacy protection. Unlike existing privacy-preserving federated learning schemes, it does not require time-consuming encryption operations or add random noise that affects model performance. It maintains the efficiency of federated learning and has the potential to be integrated with Byzantine-robust defense mechanisms.
- CPPA-SM2 is a certificateless identity-authentication scheme based on Elliptic Curve Cryptography, SM2 and the Chinese Remainder Theorem. It can verify the authenticity and integrity of the local model parameters uploaded by vehicles and supports batch verification. Unlike existing certificateless identity-authentication schemes, it integrates with the standard SM2 digital signature algorithm, facilitating practical application. Dynamic member management is achieved through the Chinese Remainder Theorem. When a malicious vehicle is detected in the system, TA can use the system master secret key to trace its real identity and then revoke it from the federated learning system.

- We conducted a security proof and an informal security analysis of the CPPA-SM2 scheme. Additionally, we evaluated its performance through experiments and compared it with other schemes. The experimental results show that CPPA-SM2 can achieve efficient and secure authentication for vehicles while providing privacy protection for federated learning.

The remainder of this paper is organized as follows. Section 2 presents the notation definitions, mathematical background, system model, threat model, security model and design objectives. Section 3 details the implementation of the CPPA-SM2 scheme. Section 4 provides the correctness and security proof of the CPPA-SM2 scheme along with an informal security analysis. Section 5 evaluates the performance of the CPPA-SM2 scheme and compares it with other schemes. Section 6 concludes the paper.

2. Preliminaries

In this section, we mainly introduce the preliminary knowledge, system model, threat model, security model and design goals. The relevant symbols used in this paper are explained in Table 1.

Table 1. Notations and definitions used.

Notations	Definition
λ	Security parameter
s	System master secret key
P_{pub}	System public key
(pk_{TA}, sk_{TA})	TA's public and private key pair
(pk_{RSU}, sk_{RSU})	RSU's public and private key pair
V_i	The i-th vehicle
K	Group key
(β, D_{pub})	Group public key
(X_i, Y_i)	Vehicle V_i's full public key
(x_i, y_i)	Vehicle V_i's full private key
sk_i	Vehicle V_i's secret key
RID_i	Vehicle V_i's real identity
$PID_i = (PID_{i,1}, PID_{i,2})$	An pseudo-identity of vehicle V_i
T_i	Current timestamp
T_a	Arrival time
ΔT	The validity period of the pseudo-identity
T_K	The validity period of the group key
H_1, H_2, H_3, H_4, H_5	Five one-way hash functions
sgk_i	The signature key for vehicle V_i
$\|\|$	Concatenation operation
SIG	Signature algorithm
W_i^t	The local model parameters of vehicle V_i in round t
$W_{RSU_j}^t$	The local model parameters aggregated by RSU_j in round t
W_{global}^{t+1}	The global model for round $t+1$

2.1. Chinese Remainder Theorem

The Chinese Remainder Theorem (CRT) [23,28] is a theorem of number theory that allows one to solve systems of simultaneous congruences with different moduli. It asserts that if one knows the remainders of the division of an integer by several pairwise coprime integers, then one can determine uniquely the remainder of the division of that integer by the product of these integers, under certain conditions.

Let $sk_1, sk_2, \ldots, sk_n$ be pairwise co-prime positive numbers and $l_1, l_2, \ldots l_n$ be any given n positive integers. Then, CRT asserts that the following simultaneous congruence equation

$$X \equiv l_1 \bmod sk_1, X \equiv l_2 \bmod sk_2, \ldots, X \equiv l_n \bmod sk_n \tag{1}$$

has a unique solution X module θ, where $\theta = sk_1 sk_2 \cdots sk_n = \prod_{i=1}^{n} sk_i$, and the X can be obtained by the following equation:

$$X = \sum_{i=1}^{n} l_i a_i b_i (\mathrm{mod}\theta), \tag{2}$$

where $a_i = \theta / sk_i$ and $b_i = (a_i)^{-1} \mathrm{mod} sk_i$.

2.2. Elliptic Curve Cryptosystem

Consider a finite field F_p determined by a prime number p. Let $E(F_p)$ be a set of elliptic curve points over F_p defined by the equation $y^2 = x^3 + ax + b \bmod p$, where $a, b \in F_p$ and $(4a^3 + 27b^2) \bmod p \neq 0$. The elliptic curve $E(F_p)$ includes both scalar multiplication and point addition operations. $\mathbb{G}$ is an additive cyclic group with order q. The Elliptic Curve Discrete Logarithm Problem (ECDLP) is defined as follows: Given two random points $P, Q \in \mathbb{G}$ on elliptic curve $E(F_p)$, where $Q = xP, x \in Z_q^*$, it has been proven that calculating x from Q is computationally difficult. In other words, it is infeasible to find x in polynomial time with a non-negligible probability [29,30].

2.3. SM2 Digital Signature Algorithm

The SM2 digital signature algorithm [31] is a public key cryptographic algorithm based on elliptic curve cryptography, developed by the Chinese State Cryptography Administration. It is part of the Chinese National Standards (GB/T 32918.1-2016) [32] and is widely used for secure communications in China. The SM2 digital signature algorithm consists of three main phases: Key Generation, Signature Generation and Signature Verification.

- Key Generation $(params) \rightarrow (d_A, P_A)$: Assume the signer of the message is user A. TA chooses the elliptic curve parameters $param = (p, a, b, q, G)$, selects a random integer $d_A \in [1, n-1]$ as the private key and calculates the public key $P_A = d_A G$ for user A.
- Signature Generation $(params, m, d_A) \rightarrow \sigma_A$: Given a message m. A computes $Z_A = H(len_{ID_A} || ID_A || a || b || G || P_A)$ and $e_A = H(Z_A || m)$, where len_{ID_A} represents two bytes converted from the bit length of user A's identity ID_A, a and b are elements in F_p that define an elliptic curve over $E(F_p)$, G denotes the base point in the elliptic curve group $\mathbb{G}$ and P_A denotes user A's public key. Then, A randomly chooses $k_A \in [1, n-1]$, calculates $\mathcal{K}_A = k_A \cdot G = (x_1, y_1)$ and $r_A = (e_A + x_1) \bmod q$. Finally A calculates $s_A = (k_A - r_A \cdot d_A)/(1 + d_A) \bmod q$, where d_A denotes user A's private key. User A's signature on the message m is $\sigma_A = (r_A, s_A)$.
- Signature Verification $(params, m, \sigma_A, P_A) \rightarrow true\ or\ false$: Assume the verifier of the signature σ_A is user B. Given user A's signature $\sigma_A = (r_A, s_A)$ on message m, if $r_A \notin [1, n-1]$ or $s_A \notin [1, n-1]$, B outputs false and exits. Then B computes $Z_A = H(len_{ID_A} || ID_A || a || b || G || P_A)$, $e_A = H(Z_A || m)$ and calculates $t_A = (r_A + s_A) \bmod q$. If $t_A = 0$, B outputs false and exits. Finally, B calculates $s_A G + t_A P_A = (x'_1, y'_1) = \mathcal{K}'_A$ and $R = (e_A + x'_1) \bmod q$. If $R = r_A$, B outputs true; otherwise, it outputs false.

2.4. System Model

In the IoV, a federated learning system primarily includes four entities: a trusted authority (TA), cloud server (CS), roadside units (RSUs) and vehicles, as shown in Figure 1.

TA: This is a trusted third party, typically the traffic-management department. It is primarily responsible for system initialization, registration of vehicles and RSUs, generating related keys for them and managing identities. In this paper, when a malicious vehicle uploads false local model parameters or forges identity information, the TA can trace its real identity and revoke it from the system.

Vehicles: These are the data owners and participants in federated learning. They use their locally collected data to train the global model received from CS, and then upload the local model parameters. In this paper, vehicles participate in federated learning using pseudonyms, sign the locally trained model parameters and then send them to the nearby RSU.

RSUs: These verify the authenticity and integrity of the local model parameters uploaded by vehicles. They use the FedAvg algorithm [5] to perform local aggregation on these parameters to obtain local aggregation results, which are then uploaded to the cloud server for global aggregation. Additionally, they broadcast the global model issued by TA to the vehicles within their communication range.

CS: Upon receiving the local aggregation results uploaded by RSUs, CS uses FedAvg to perform global aggregation to obtain the global model for the next round of training. The new global model is then distributed to the vehicles to begin the next training round. Through multiple iterations, the performance of the global model can be improved, enabling the cloud server to utilize the results for practical predictions, judgments and applications.

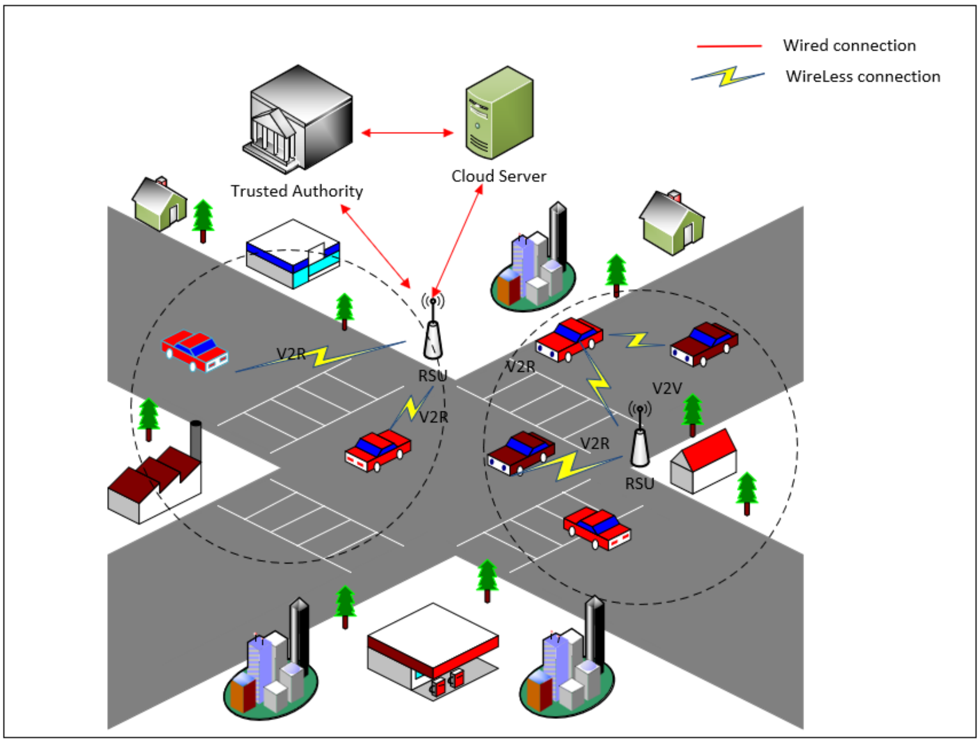

Figure 1. Authentication scheme based on CPPA-SM2 for IoV.

2.5. Threat Model and Security Model

In the threat model, CS and RSUs are considered honest-but-curious. This means they will honestly follow the protocol to verify vehicle identities and the authenticity and integrity of model parameters, and they will aggregate local models to obtain the global model [33]. However, they are curious about the private data owned by the vehicles and may attempt to recover the vehicles' original data and reveal their true identities by analyzing the received model parameters. Therefore, they might pose a threat to vehicle privacy. Vehicles may be malicious and can launch free-riding attacks and data-poisoning attacks by uploading false model parameters. They may also forge identities and signatures to attempt to have fake messages successfully authenticated by RSUs. Additionally, they might try to infer the privacy information of other vehicles. Attackers can fully control the wireless communication channels between vehicles, RSUs, TA and CS. They can intercept messages on the channel, tamper with messages, replay old messages and attempt to impersonate other vehicles to send messages [34].

Based on the aforementioned threats and the certificateless signature security model [27,28,30], our proposed security model is as follows. The hash functions used in this model are assumed to be random oracles.

In the security model, we consider two types of adversaries, $\mathcal{A}_I$ and $\mathcal{A}_{II}$. $\mathcal{A}_I$ can launch public key-replacement attacks but cannot access system master secret key s. $\mathcal{A}_{II}$ can access the system master secret key but cannot perform public key-replacement attacks. Both types of adversaries will engage in two separate games with the challenger $\mathcal{C}$.

Game 1: This security game is executed between $\mathcal{A}_I$ and $\mathcal{C}$. $\mathcal{C}$ initializes the system using the security parameter λ generating system master secret key s and system public parameters $param$. $\mathcal{C}$ secretly keeps s and sends the public parameters to $\mathcal{A}_I$. $\mathcal{A}_I$ can perform the following queries.

- Hash queries: Upon receiving a query from $\mathcal{A}_I$, $\mathcal{C}$ returns the corresponding hash values to $\mathcal{A}_I$.
- Partial-Private-Key-Extract-queries: Upon receiving a query with a pseudonym PID_i, $\mathcal{C}$ returns the partial private key y_i of the vehicle to $\mathcal{A}_I$.
- Public-Key-Extract-queries: Upon receiving a query with a pseudonym PID_i, $\mathcal{C}$ returns the public key (X_i, Y_i) of the vehicle to $\mathcal{A}_I$.
- Secret-Value-Extract-queries: Upon receiving a query with a pseudonym PID_i, $\mathcal{C}$ returns the secret value x_i of the vehicle to $\mathcal{A}_I$.
- Public-Key-Replace-queries: Upon receiving a query with $(PID_i, (X_i', Y_i'))$, $\mathcal{C}$ replaces public key with the new public key (X_i', Y_i').
- Sign queries: After receiving a query from $\mathcal{A}_I$ with $\{PID_{i,1}, PID_{i,2}, M_i, T_i\}$, $\mathcal{C}$ responds with a signature σ_i.
- Forgery: Once $\mathcal{A}_I$ has completed the desired queries, it outputs $\left\{M_i^*, PID_{i,1}^*, PID_{i,2}^*, T_i^*, \sigma_i^*\right\}$ under the pseudo identity PID^*_{i}. $\mathcal{A}_I$ wins the game if the following conditions are met:
- σ_i^* passes verification.
- Partial-Private-Key-Extract-queries oracle has not received the request with PID_i^*.
- Sign queries oracle has not received the request with $\left\{M_i^*, PID_{i,1}^*, PID_{i,2}^*, T_i^*\right\}$.

Definition 1. *CPPA-SM2 is existentially unforgeable under adaptive chosen-identity and chosen-message attacks if no polynomial-time adversary $\mathcal{A}_I$ can win the above game with non-negligible advantage.*

Game 2: This security game is executed between $\mathcal{A}_{II}$ and $\mathcal{C}$. $\mathcal{C}$ initializes the system using the security parameter λ generating system master secret key s and system public parameters $param$. $\mathcal{C}$ sends them to $\mathcal{A}_{II}$.

- Query: $\mathcal{A}_{II}$ can perform all the queries from Game 1 except for Public-Key-Replace-queries.
- Forgery: Once $\mathcal{A}_{II}$ has completed the desired queries, it outputs $\left\{M_i^*, PID_{i,1}^*, PID_{i,2}^*, T_i^*, \sigma_i^*\right\}$ under the pseudo identity PID^*_{i}. $\mathcal{A}_{II}$ wins the game if the following conditions are met:
- σ_i^* passes verification.
- Secret-Value-Extract-queries oracle has not received the request with PID_i^*.
- Sign queries oracle has not received the request with $\left\{M_i^*, PID_{i,1}^*, PID_{i,2}^*, T_i^*\right\}$.

Definition 2. *CPPA-SM2 is existentially unforgeable under adaptive chosen-identity and chosen-message attacks if no polynomial-time adversary $\mathcal{A}_{II}$ can win the above game with non-negligible advantage.*

2.6. Design Goals

Under the security model, CPPA-SM2 primarily has the following design goals:

Anonymity and Privacy-Preserving: CPPA-SM2 should protect the privacy of vehicles participating in federated learning training. No entity other than TA should be able to infer the true identity of the vehicles.

Authenticity and Integrity: CPPA-SM2 should ensure that the local model parameters received by RSUs are from legitimate vehicles and that they have not been tampered with during transmission.

Un-linkability: Attackers cannot link any two messages sent by the same vehicle.

Un-forgeability: Attackers cannot forge signatures of other vehicles on messages, allowing RSUs to successfully verify the signatures.

Non-repudiation: Once a vehicle uploads local model parameters and they are authenticated, the vehicle cannot deny its contribution to the global model.

Forward Security: When a vehicle joins a group, it cannot access communications that occurred before its joining, meaning it cannot participate in previous federated learning training processes of the group.

Backward Security: When a vehicle leaves the group or is revoked by the TA, it cannot participate in the current model training process or access communications that occur after its departure from the group.

In addition to achieving the aforementioned security goals, CPPA-SM2 should also have efficient authentication efficiency and lower communication overhead to adapt to the communication environment of IoV. In particular, when a large number of vehicles participate in federated learning training, RSUs should be able to authenticate them in batches.

3. The Proposed Scheme

In this section, we present a certificateless conditional privacy-preserving identity-authentication protocol based on CRT and the SM2 digital signature algorithm, named CPPA-SM2. CPPA-SM2 aims to provide privacy protection for vehicles participating in federated learning. It consists of five phases: system initialization, registration, message sign, message verification and group member management. First, TA initializes the system and publishes the system's public parameters. Then, vehicles and RSUs register with TA before participating in communications. Through registration, they obtain the public and private keys required for subsequent communications. In the message signing phase, vehicles train a model based on their local datasets and then sign the local model parameters before sending them to RSU. RSU, upon receiving the local model parameters from nearby vehicles, verifies the signatures and aggregates the verified local model parameters to obtain a local aggregation result. RSU then sends this local aggregation result to CS for global aggregation, resulting in the next round of the global model. If a malicious vehicle is detected uploading malicious model parameters or forging signatures, TA can trace its identity and revoke it from the system. The overall workflow of CPPA-SM2 is illustrated in Figure 2 and Protocol 1. The details of the scheme are as follows.

Protocol 1 CPPA-SM2

```
① System Initialization
For TA:
1: Use $\lambda$ to generate two large prime numbers $p$ and $q$.
2: Randomly select $s \in Z_q^*$ and calculates $P_{pub} = s \cdot G$.
3: Choose five one-way hash functions $H_i = \{0,1\}^* \to Z_q^*, i = 1,2,3,4,5$.
4: Publish $param = \left\{p, q, E(F_p), G, \mathbb{G}, Z_q^*, P_{pub}, H_1, H_2, H_3, H_4, H_5\right\}$.
② Registration
For each vehicle:
1: $V_i$ randomly selects $x_i \in Z_q^*$, calculates $X_i = x_i \cdot G$ and send $(RID_i, X_i)$ to TA.
2: Upon receiving $(RID_i, X_i)$, TA calculates $h_i = H_1(X_i||P_{pub})$, $y_i = s \cdot h_i$, $Y_i = y_i \cdot G$ and randomly selects $sk_i \in Z_q^*$. Then, TA sends $y_i$, $Y_i$ and $sk_i$ to $V_i$.
3: $V_i$ sets $(X_i, Y_i)$, $(x_i, y_i)$ and $sk_i$.
```

Protocol 1 CPPA-SM2

```
For each RSU:
1: $RSU_j$ sends $ID_{RSUj}$ to TA.
2: TA generates a pair of public and private keys $(sk_{RSU_j}, pk_{RSU_j})$ and sends them to $RSU_j$.
3. $RSU_j$ sets $(sk_{RSU_j}, pk_{RSU_j})$.
For TA:
1: Calculate $\theta = \prod_{i=1}^{n} sk_i$, $a_i = \theta / sk_i$, $b_i = (a_i)^{-1}$ mod $sk_i$ and set $c_i = a_i \cdot b_i$, $u = \sum_{i=1}^{n} c_i$.
2: Randomly pick a group key $K \in Z_q^*$ and calculate the group public key $\beta = K \cdot u$ and $D_{pub} = K \cdot G$.
3: Sign $\beta$, $D_{pub}$ and the $K$'s valid period $T_K$ using its private key $sk_{TA}$ and broadcast the information $\left\{\beta, D_{pub}, SIG_{sk_{TA}}(\beta || D_{pub} || T_K)\right\}$ to vehicles and RSUs in $C_n$.
③ Message Sign
For each vehicle:
1: $V_i$ trains the global model $W_{global}^t$ using its local dataset $D_i$ to obtain the local model parameters $W_i^t$.
2: $V_i$ randomly selects $c_i \in Z_q^*$ to generate a pseudo identity $PID_i = (PID_{i,1}, PID_{i,2})$, where $PID_{i,1} = c_i \cdot G$ and $PID_{i,2} = RID_i \oplus H_2(c_i \cdot P_{pub})$.
3: $V_i$ calculates $Z_i = H_3(len_{PID_{i,2}} || PID_{i,2} || a || b || G || X_i)$, $\varphi_i = H_4(PID_{i,1} || T_i)$ and $sgk_i = y_i + Z_i \cdot K + x_i \cdot \varphi_i$.
4: $V_i$ randomly selects $k_i \in Z_q^*$, calculates $\mathcal{K}_i = k_i \cdot G = (x_1, y_1)$ $e_i = H_5(Z_i || W_i^t | T_i)$, $r_i = e_i + x_1$ mod $q$ and $s_i = (1 + sgk_i)^{-1} \cdot (k_i - r_i \cdot sgk_i)$ mod $q$.
5. $V_i$ obtains the signature $\sigma^t_{\ i} = (r_i, s_i)$ of $W_i^t$ and sends messages $\left\{W^t_{\ i}, \sigma^t_{\ i}, (X_i, Y_i), PID_i, T_i\right\}$ to the nearby $RSU_j$.
④ Message Verification
For each RSU:
1: Upon receiving the messages $\left\{W^t_{\ i}, \sigma^t_{\ i}, (X_i, Y_i), PID_i, T_i\right\}$ from $V_i$, $RSU_j$ first checks the validity of timestamp. If $\Delta T \geq T_a - T_i$, where $T_a$ represents the arrival time, continues; otherwise, discards.
2: $RSU_j$ calculates $Z_i = H_3(len_{PID_{i,2}} || PID_{i,2} || a || b || G || X_i)$, $e_i = H_5(Z_i || W_i^t || T_i)$, $\varphi_i = H_4(PID_{i,1} || T_i)$, $t_i = r_i + s_i$ mod $q$ and $\mathcal{K}'_{\ i} = (x'_1, y'_1) = s_i \cdot G + t_i \cdot [Y_i + Z_i \cdot D_{pub} + \varphi_i \cdot X_i]$.
3: $RSU_j$ checks the equality of $R = e_i + x'_1 = r_i$ for authentication and validity.
4: $RSU_j$ uses the FedAvg algorithm to locally aggregate the verified local model parameters $\{W^t_{\ 1}, W^t_{\ 2}, \ldots, W^t_{\ n}\}$, producing a local aggregation result $W_{RSU_j}^t \leftarrow FedAvg(W_i^t, n)$.
5: $RSU_j$ signs this result with its private key and sends messages $\{W_{RSU_j}^t, SIG_{sk_{RSU_j}}(W_{RSU_j}^t)\}$ to CS.
For CS:
1: CS performs a global aggregation on the verified local aggregation results $\{W_{RSU_1}^t, W_{RSU_2}^t, \ldots, W_{RSU_m}^t\}$ to obtain the global model $W_{global}^{t+1} \leftarrow FedAvg(W_{RSU_j}^t, m)$.
2: CS signs the global model with its private key and sends messages $\{W_{global}^{t+1}, SIG_{sk_{CS}}(W_{global}^{t+1})\}$ to the vehicles within the communication group via RSUs.
⑤ Group Member Management
Trace:
1: TA uses the system's master private key $s$ to recover the vehicle's true identity $RID_i = PID_{i,2} \oplus H_2(s \cdot PID_{i,1})$.
Revoke:
1. TA first removes $c_i$ related to $V_i$ from $u$ by computing $u' = u - c_i$.
2: TA randomly selects a new group key $K' \in Z_q^*$, calculates new group public keys $\beta' = K' \cdot u'$ and $D'_{pub} = K' \cdot G$, and broadcasts the updated information $\{\beta', D'_{pub}, SIG_{sk_{TA}}(\beta' || D'_{pub} || T_{K'})\}$ to vehicles and RSUs in $C_n$.
Add:
1. TA randomly selects a new group key $K' \in Z_q^*$ and calculates $\theta' = \theta \cdot f_i$, $a'_i = \theta' / f_i$, $b'_i = (a'_i)^{-1}$ mod $sk_i$, $c'_i = a'_{\ i} \cdot b'_{\ i}$ and $u' = \sum_{i=1}^{n} c'_i$.
2. TA computes new group public keys $\beta' = K' \cdot u'$ and $D'_{pub} = K' \cdot G$, and broadcasts the updated information $\{\beta', D'_{pub}, SIG_{sk_{TA}}(\beta' || D'_{pub} || T_{K'})\}$ in $C_n$.
```

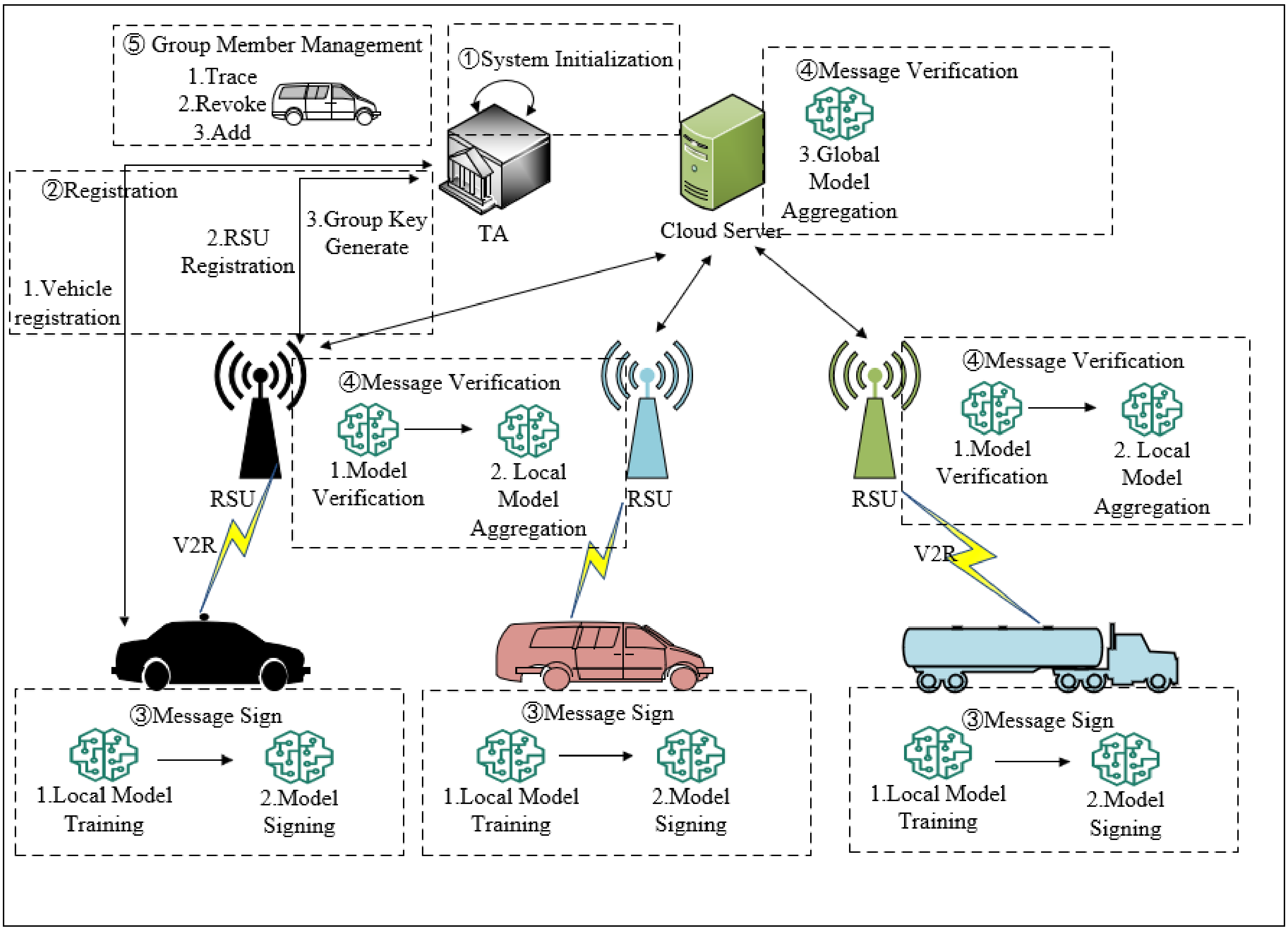

Figure 2. Workflow of CPPA-SM2.

3.1. System Initialization

TA uses a security parameter λ to generate two large prime numbers p and q, where $p > q$, $q \leq \lceil p/4 \rceil$. Let $E(F_p)$ denote an elliptic curve over the finite field F_p and G denote a base point on the elliptic curve $E(F_p)$ with order q. Let $\mathbb{G}$ be an additive cyclic group generated by G. TA randomly selects $s \in Z_q^*$ as the system master secret key and calculates the system public key $P_{pub} = s \cdot G$. Then, TA chooses five one-way hash functions $H_i = \{0,1\}^* \to Z_q^*, i = 1,2,3,4,5$. TA secretly holds s and publishes the system's public parameters $param = \left\{p, q, E(F_p), G, \mathbb{G}, Z_q^*, P_{pub}, H_1, H_2, H_3, H_4, H_5\right\}$.

3.2. Registration

In the registration phase, both vehicles and RSUs need to register with TA to obtain the relevant keys for subsequent communications. We assume that TA is fully trusted and that the entire registration phase is conducted over a secure channel, eliminating the risk of privacy leaks and security attacks.

3.2.1. Vehicle Registration

For a vehicle V_i with its real identity RID_i, it first randomly selects $x_i \in Z_q^*$ as its secret value and calculates $X_i = x_i \cdot G$ as its first part of the public key. Then, V_i sends (RID_i, X_i) to TA. Upon receiving (RID_i, X_i), TA calculates $h_i = H_1(X_i || P_{pub})$, $y_i = s \cdot h_i$ and $Y_i = y_i \cdot G$, where y_i and Y_i serve as V_i's partial private key and the second part of the public key. In addition, TA randomly selects a prime number $sk_i \in Z_q^*$ as a secret key for V_i. Completing these computations, TA returns y_i, Y_i and sk_i to V_i. Upon receiving y_i, Y_i

and sk_i, V_i sets (x_i, y_i) as its full private key, (X_i, Y_i) as its full public key and uses sk_i for subsequent group communications.

3.2.2. RSU Registration

For a roadside unit RSU_j with its identity ID_{RSUj}, TA generates a pair of public and private keys (sk_{RSU_j}, pk_{RSU_j}). Then, TA distributes them to RSU_j. Here, we assume that all vehicles know the public keys of TA and RSUs.

3.2.3. Group Key Generate

To ensure that the uploaded local model parameters come from legitimate vehicles and to support efficient group communication, TA constructs a communication group C_n for them based on the secret keys sk_i of n vehicles and CRT. TA first calculates $\theta = \prod_{i=1}^{n} sk_i$, $a_i = \theta / sk_i$ and $b_i = (a_i)^{-1} \bmod sk_i$. TA sets $c_i = a_i \cdot b_i$, $u = \sum\limits_{i=1}^{n} c_i$, where $i = 1, 2, \ldots, n$. Then, TA randomly picks a group key $K \in Z_q^*$ and calculates the group public key $\beta = K \cdot u$ and $D_{pub} = K \cdot G$. TA signs β, D_{pub} and the K's valid period T_K using its private key sk_{TA} and broadcasts the information $\{\beta, D_{pub}, SIG_{sk_{TA}}(\beta || D_{pub} || T_K)\}$ to vehicles and RSUs in C_n. Once receiving the broadcast information, any authorized vehicle in C_n can obtain K by performing a modulus operation $K \equiv \beta \bmod sk_i$ according to CRT.

3.3. Message Sign

In the $t - th$ round of training, the vehicle V_i trains the global model W_{global}^t using its local dataset D_i to obtain the local model parameters W_i^t, i.e., $W_i^t \leftarrow W_{global}^t - \eta \nabla \mathcal{L}(W_{global}^t, D_i)$. Before sending the local model parameter W_i^t to the nearby RSU_j, the vehicle V_i signs it as follows to ensure the authenticity and integrity of W_i^t.

V_i randomly selects $c_i \in Z_q^*$ to generate a pseudo identity $PID_i = (PID_{i,1}, PID_{i,2})$, where $PID_{i,1} = c_i \cdot G$ and $PID_{i,2} = RID_i \oplus H_2(c_i \cdot P_{pub})$. Then, V_i calculates $Z_i = H_3$ $(len_{PID_{i,2}} || PID_{i,2} || a || b || G || X_i)$, $\varphi_i = H_4(PID_{i,1} || T_i)$ and signature key $sgk_i = y_i + Z_i \cdot K + x_i \cdot \varphi_i$, where $len_{PID_{i,2}}$ represents two bytes converted from the bit length of $PID_{i,2}$, a and b are elements in F_p that define an elliptic curve over $E(F_p)$ and T_i represents the current timestamp. Next, V_i randomly selects $k_i \in Z_q^*$ and calculates $\mathcal{K}_i = k_i \cdot G = (x_1, y_1)$ $e_i = H_5(Z_i || W_i^t | T_i)$, $r_i = e_i + x_1 \bmod q$ and $s_i = (1 + sgk_i)^{-1} \cdot (k_i - r_i \cdot sgk_i) \bmod q$. For simplicity, we omit the notation t of PID_i, Z_i, φ_i, sgk_i, $\mathcal{K}_i$, e_i, r_i and s_i. Finally, V_i obtains the signature $\sigma^t_{\ i} = (r_i, s_i)$ of W_i^t and sends messages $\{W^t_{\ i}, \sigma^t_{\ i}, (X_i, Y_i), PID_i, T_i\}$ to the nearby RSU_j.

3.4. Message Verification

3.4.1. Single Message Verification

Upon receiving the messages $\{W^t_{\ i}, \sigma^t_{\ i}, (X_i, Y_i), PID_i, T_i\}$ from V_i, RSU_j first checks the validity of the timestamp. If $\Delta T \geq T_a - T_i$, where T_a represents the arrival time, it continues; otherwise, it discards. Then RSU_j calculates $Z_i = H_3(len_{PID_{i,2}} || PID_{i,2} || a || b || G || X_i)$, $e_i = H_5(Z_i || W_i^t || T_i)$, $\varphi_i = H_4(PID_{i,1} || T_i)$, $t_i = r_i + s_i \bmod q$ and $\mathcal{K}'_{\ i} = (x_1', y_1') = s_i \cdot G + t_i \cdot [Y_i + Z_i \cdot D_{pub} + \varphi_i \cdot X_i]$. Finally, RSU_j checks the equality of $R = e_i + x_1' = r_i$ for authentication and validity.

3.4.2. Batch Messages Verification

When receiving a batch of messages $\{W_1^t, \sigma_1^t, (X_1, Y_1), PID_1, T_1\}$, $\{W_2^t, \sigma_2^t, (X_2, Y_2), PID_2, T_2\}$, ..., $\{W_n^t, \sigma_n^t, (X_n, Y_n), PID_n, T_n\}$ from the vehicles $\{V_1, V_2, \ldots, V_n\}$, RSU_j first checks the validity of timestamp T_i, where $i = 1, 2, \ldots, n$. If T_i is valid, it continues; otherwise, it discards. To prevent confusion attacks while ensuring non-repudiation, CPPA-SM2 uses a set of small exponents $\{v_1, v_2, \ldots, v_n\}$ for batch verification [23,35], where $v_i \in [1, 2^t]$ and t is a small integer. Next, RSU_j calculates

$$(x'_1, y'_1) = \sum_{i=1}^{n} (v_i \cdot s_i) \cdot G + \sum_{i=1}^{n} (v_i \cdot t_i \cdot Y_i) + \sum_{i=1}^{n} (v_i \cdot t_i \cdot Z_i) \cdot D_{pub} + \sum_{i=1}^{n} (v_i \cdot t_i \cdot \varphi_i \cdot X_i), \quad (3)$$

and checks whether $R = \sum_{i=1}^{n} (v_i \cdot e_i) + x'_1 = \sum_{i=1}^{n} (v_i \cdot r_i)$ holds or not. If true, all messages are valid; otherwise, some of these messages are invalid. The detection algorithm for invalid message signatures has been proposed in [36]. The details of this algorithm are beyond the scope of this paper.

3.4.3. Local Model Aggregation

RSU_j uses the FedAvg algorithm to locally aggregate the verified local model parameters $\{W^t_{\ 1}, W^t_{\ 2}, \ldots, W^t_{\ n}\}$, producing a local aggregation result $W^t_{RSU_j} \leftarrow FedAvg(W^t_i, n)$, where $i \in [1, n]$ and n denotes the number of vehicles participating in the training within the RSU_j's range. It then signs this result with its private key and sends messages $\{W^t_{RSU_j}, SIG_{sk_{RSU_j}}(W^t_{RSU_j})\}$ to CS. Upon receiving the local aggregation result $W^t_{RSU_j}$ from RSUs, CS verifies its validity. It then performs a global aggregation on the verified local aggregation results $\left\{W^t_{RSU_1}, W^t_{RSU_2}, \ldots, W^t_{RSU_m}\right\}$ to obtain the global model $W^{t+1}_{global} \leftarrow FedAvg(W^t_{RSU_j}, m)$, where $j \in [1, m]$ and m denotes the number of RSUs. CS signs the global model with its private key and sends messages $\{W^{t+1}_{global}, SIG_{sk_{TA}}(W^{t+1}_{global})\}$ to the vehicles within the communication group via RSUs.

3.5. Group Member Management

3.5.1. Trace

When RSU_j detects that a vehicle V_i has uploaded malicious local model parameters or has engaged in identity forgery, it sends the vehicle's pseudonym PID_i to TA. TA then uses the system's master private key s to recover the vehicle's true identity $RID_i = PID_{i,2} \oplus H_2(s \cdot PID_{i,1})$.

3.5.2. Revoke

Upon obtaining the true identity RID_i of the malicious vehicle V_i, TA can completely remove it from the federated learning system by revoking its legitimate information from the group. TA first removes c_i related to V_i from u by computing $u' = u - c_i$. Then, TA randomly selects a new group key $K' \in Z_q^*$, calculates new group public keys $\beta' = K' \cdot u'$ and $D'_{pub} = K' \cdot G$ and broadcasts the updated information $\{\beta', D'_{pub}, SIG_{sk_{TA}}(\beta'||D'_{pub}||T_{K'})\}$ to vehicles and RSUs in C_n. Upon receiving $\{\beta', D'_{pub}, SIG_{sk_{TA}}(\beta'||D'_{pub}||T_{K'})\}$, the remaining vehicles in C_n can use their secret key sk_j to compute the updated group key $K' = \beta' \bmod sk_j$. Since u' no longer contains the legitimate information of V_i, it cannot compute the new group key K'. When a vehicle leaves the communication group C_n, TA can also revoke it in this way.

3.5.3. Add

When a vehicle V_i applies to join the federated learning system, TA randomly selects a new group key $K' \in Z_q^*$ and calculates $\theta' = \theta \cdot f_i$, $a'_i = \theta' / f_i$, $b'_i = (a'_i)^{-1} \bmod sk_i$, $c'_i = a'_{\ i} \cdot b'_{\ i}$ and $u' = \sum_{i=1}^{n} c'_i$. Then, TA computes new group public keys $\beta' = K' \cdot u'$ and $D'_{pub} = K' \cdot G$, and broadcasts the updated information $\{\beta', D'_{pub}, SIG_{sk_{TA}}(\beta'||D'_{pub}||T_{K'})\}$ in C_n. Upon receiving $\{\beta', D'_{pub}, SIG_{sk_{TA}}(\beta'||D'_{pub}||T_{K'})\}$, vehicles in C_n, it calculates the updated group key $K' = \beta' \mathrm{mod} sk_i$.

4. Correctness and Security Proof and Analysis

In this section, we first provide a proof of correctness for the proposed scheme. Then, under the random oracle model, we prove the security of the scheme. Finally, we conduct an informal security analysis of the scheme.

4.1. Correctness Proof

The correctness verification of the single message signature is ensured by Equations (4) and (5).

$$\begin{aligned} \mathcal{K}'_{i} = (x'_1, y'_1) &= s_i \cdot G + t_i \cdot [Y_i + Z_i \cdot D_{pub} + \varphi_i \cdot X_i] \\ &= s_i \cdot G + (r_i + s_i)[Y_i + Z_i \cdot D_{pub} + \varphi_i \cdot X_i] \\ &= s_i \cdot G + r_i \cdot [Y_i + Z_i \cdot D_{pub} + \varphi_i \cdot X_i] + s_i \cdot [Y_i + Z_i \cdot D_{pub} + \varphi_i \cdot X_i] \\ &= s_i \cdot G(1 + y_i + Z_i \cdot K + \varphi_i \cdot x_i) + r_i \cdot G(y_i + Z_i \cdot K + \varphi_i \cdot x_i) \\ &= (1 + sgk_i)^{-1} \cdot (k_i - r_i \cdot sgk_i) \cdot G \cdot (1 + sgk_i) + r_i \cdot G \cdot (sgk_i) \\ &= (1 + sgk_i)^{-1} \cdot k_i \cdot G \cdot (1 + sgk_i) - (1 + sgk_i)^{-1} \cdot r_i \cdot sgk_i \cdot G \cdot (1 + sgk_i) + r_i \cdot G \cdot (sgk_i) \\ &= k_i \cdot G - r_i \cdot sgk_i \cdot G + r_i \cdot G \cdot (sgk_i) \\ &= k_i \cdot G \\ &= \mathcal{K}_i = (x_1, y_1) \end{aligned} \tag{4}$$

$$R = e_i + x'_1 = r_i = e_i + x_1 \tag{5}$$

The correctness verification of the batch message signatures is ensured by Equations (6) and (7).

$$\begin{aligned} &\left(\sum_{i=1}^{n} v_i \cdot \mathcal{K}'_{i}\right) = (x'_1, y'_1) = \left(\sum_{i=1}^{n} v_i \cdot s_i\right) \cdot G + \left(\sum_{i=1}^{n} v_i \cdot t_i \cdot [Y_i + Z_i \cdot D_{pub} + \varphi_i \cdot X_i]\right) \\ &= \left(\sum_{i=1}^{n} v_i \cdot s_i\right) \cdot G + \left(\sum_{i=1}^{n} v_i \cdot (r_i + s_i) \cdot [Y_i + Z_i \cdot D_{pub} + \varphi_i \cdot X_i]\right) \\ &= \left(\sum_{i=1}^{n} v_i \cdot s_i\right) \cdot G + \left(\sum_{i=1}^{n} v_i \cdot r_i \cdot [Y_i + Z_i \cdot D_{pub} + \varphi_i \cdot X_i]\right) + \left(\sum_{i=1}^{n} v_i \cdot s_i \cdot [Y_i + Z_i \cdot D_{pub} + \varphi_i \cdot X_i]\right) \\ &= \left(\sum_{i=1}^{n} v_i \cdot s_i \cdot G(1 + y_i + Z_i \cdot K + \varphi_i \cdot x_i)\right) + \left(\sum_{i=1}^{n} v_i \cdot r_i \cdot G(y_i + Z_i \cdot K + \varphi_i \cdot x_i)\right) \\ &= \left(\sum_{i=1}^{n} v_i \cdot (1 + sgk_i)^{-1} \cdot (k_i - r_i \cdot sgk_i) \cdot G \cdot (1 + sgk_i)\right) + \left(\sum_{i=1}^{n} v_i \cdot r_i \cdot G \cdot (sgk_i)\right) \\ &= \left(\sum_{i=1}^{n} v_i \cdot k_i \cdot G\right) - \left(\sum_{i=1}^{n} v_i \cdot r_i \cdot sgk_i \cdot G\right) + \left(\sum_{i=1}^{n} v_i \cdot r_i \cdot G \cdot (sgk_i)\right) \\ &= \left(\sum_{i=1}^{n} v_i \cdot k_i \cdot G\right) \\ &= \left(\sum_{i=1}^{n} v_i \cdot \mathcal{K}_i\right) = (x_1, y_1) \end{aligned} \tag{6}$$

$$R = \left(\sum_{i=1}^{n} v_i \cdot e_i\right) + x'_1 = \left(\sum_{i=1}^{n} v_i \cdot r_i\right) = \left(\sum_{i=1}^{n} v_i \cdot e_i\right) + x_1 \tag{7}$$

Based on the signing and verification process, if the local model parameter W_i^t and signature $\sigma^{t}_{i} = (r_i, s_i)$ transmitted by the vehicle V_i have not been tampered with and the signature $\sigma^{t}_{i} = (r_i, s_i)$ is generated using the legitimate vehicle's private key, then according to (4)–(7), RSU can correctly compute that $\mathcal{K}_i = k_i \cdot G = (x_1, y_1) = \mathcal{K}'_{i}$, thereby making $R = e_i + x'_1 = r_i = e_i + x_1$.

The correctness of legitimate vehicles in C_n obtaining the correct group key K is ensured by Equation (8).

$$\begin{aligned}&\beta(\text{mod } sk_i)\\&= K \cdot u(\text{mod } sk_i)\\&= K \cdot (a_1 \cdot b_1 + \ldots + a_n \cdot b_n)(\text{mod } sk_i)\\&= K \cdot a_i \cdot b_i(\text{mod } sk_i)\\&= K\end{aligned} \tag{8}$$

When vehicle V_i is revoked from the group C_n by TA, since $u' = u - c_i = (a_1 \cdot b_1 + \ldots + a_n \cdot b_n) - a_i \cdot b_i$, the revoked vehicle will be unable to obtain the correct group key according to Equation (9).

$$\begin{aligned}&\beta'(\text{mod } sk_i)\\&= K' \cdot u'(\text{mod } sk_i)\\&= K' \cdot (a_1 \cdot b_1 + \ldots + a_n \cdot b_n - a_i \cdot b_i)(\text{mod } sk_i)\\&\neq K'\end{aligned} \tag{9}$$

4.2. Security Proof

The security of CPPA-SM2 relies on the ECDLP. In the random oracle model, if there exist adversaries $\mathcal{A}_I$ and $\mathcal{A}_{II}$ who can win games 1 and 2 with non-negligible probabilities, respectively, then there exists a probabilistic polynomial-time simulator that can solve the ECDLP with non-negligible probability.

Theorem 1. *CPPA-SM2 is existentially unforgeable under adaptive chosen-identity and chosen-message attacks against $\mathcal{A}_I$ with the assumption that ECDLP is hard to resolve.*

Proof of Theorem 1. Let $\mathcal{C}$ be the solver of the ECDLP. Suppose that $\mathcal{A}_I$ can succeed in forging a valid signature by interacting with $\mathcal{C}$. $\mathcal{C}$ utilizes $\mathcal{A}_I$ to solve the ECDLP. Here, we give an ECDLP instance $\{G, G' = g \cdot G\}$. $\mathcal{C}$ executes the simulation to compute g through interacting with $\mathcal{A}_I$ as follows.

- Setup: On input $\{G, G'\}$, $\mathcal{C}$ sets $P_{pub} = G'$ and returns $\{p, q, E(F_p), G, Z_q^*, P_{pub}, H_1, H_2, H_3, H_4, H_5\}$ to $\mathcal{A}_I$. $\mathcal{A}_I$ selects $PID_i = (PID_{i,1}, PID_{i,2})$ as a target vehicle. In addition, $\mathcal{C}$ maintains five lists $L = \{PID_{i,1}, PID_{i,2}, x_i, y_i, X_i, Y_i\}$, $L_{H_1} = \{h_i, X_i, P_{pub}\}$, $L_{H_3} = \{Z_i, len(PID_{i,2}), PID_{i,2}, a, b, G, X_i\}$, $L_{H_4} = \{\varphi_i, PID_{i,1}, T_i\}$, $L_{H_5} = \{e_i, Z_i, M_i, T_i\}$, which are empty initially.
- Query: $\mathcal{A}_I$ can adaptively make the following queries:
- H_1-queries: After receiving the queries from $\mathcal{A}_I$ with $\{X_i, P_{pub}\}$, $\mathcal{C}$ checks whether $\{X_i, P_{pub}\}$ exists in L_{H_1}. If it does, $\mathcal{C}$ returns h_i to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ selects $h_i \in Z_q^*$ randomly and adds $\{h_i, X_i, P_{pub}\}$ to L_{H_1}. Then, $\mathcal{C}$ returns h_i to $\mathcal{A}_I$.
- H_3-queries: When receiving the queries with $\{len(PID_{i,2}), PID_{i,2}, a, b, G, X_i\}$ from $\mathcal{A}_I$, $\mathcal{C}$ checks whether $\{len(PID_{i,2}), PID_{i,2}, a, b, G, X_i\}$ exists in L_{H_3}. If it does, $\mathcal{C}$ returns Z_i to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ selects $Z_i \in Z_q^*$ randomly and adds $\{Z_i, len(PID_{i,2}), PID_{i,2}, a, b, G, X_i\}$ to L_{H_3}. Then, $\mathcal{C}$ returns Z_i to $\mathcal{A}_I$.
- H_4-queries: Upon receiving the queries from $\mathcal{A}_I$ with $\{PID_{i,1}, T_i\}$, $\mathcal{C}$ checks whether $\{PID_{i,1}, T_i\}$ exists in L_{H_4}. If it does, $\mathcal{C}$ returns φ_i to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ selects $\varphi_i \in Z_q^*$ randomly and adds $\{\varphi_i, PID_{i,1}, T_i\}$ to L_{H_4}. Then, $\mathcal{C}$ returns φ_i to $\mathcal{A}_I$.
- H_5-queries: Upon receiving the queries from $\mathcal{A}_I$ with $\{Z_i, M_i, T_i\}$, $\mathcal{C}$ checks whether $\{Z_i, M_i, T_i\}$ exists in L_{H_5}. If it does, $\mathcal{C}$ returns e_i to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ selects $e_i \in Z_q^*$ randomly and adds $\{e_i, Z_i, M_i, T_i\}$ to L_{H_5}. Then, $\mathcal{C}$ returns e_i to $\mathcal{A}_I$.
- Partial-Private-Key-Extract-queries: After receiving the queries from $\mathcal{A}_I$ with $PID_i = (PID_{i,1}, PID_{i,2})$, $\mathcal{C}$ checks whether $\{PID_{i,1}, PID_{i,2}, x_i, y_i, X_i, Y_i\}$ exists in L. If it does, $\mathcal{C}$ returns y_i to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ selects $h_i \in Z_q^*$ randomly, computes $y_i = s \cdot h_i$, $Y_i = y_i \cdot G$. Then, $\mathcal{C}$ sets $x_i = X_i = \perp$. After that, $\mathcal{C}$ adds $\{PID_{i,1}, PID_{i,2}, x_i, y_i, X_i, Y_i\}$ into L and returns y_i to $\mathcal{A}_I$.
- Public-Key-Extract-queries: After receiving the queries from $\mathcal{A}_I$ with $PID_i = (PID_{i,1}, PID_{i,2})$, $\mathcal{C}$ checks whether $\{PID_{i,1}, PID_{i,2}, x_i, y_i, X_i, Y_i\}$ exists in L. If it does, $\mathcal{C}$ returns (X_i, Y_i) to $\mathcal{A}_I$.

Otherwise, $\mathcal{C}$ does the Partial-Private-Key-Extract-queries to obtain y_i. Then, $\mathcal{C}$ selects $x \in Z_q^*$ randomly and computes $X_i = x \cdot G$, $x_i = x$, $Y_i = y_i \cdot G$. After that, $\mathcal{C}$ adds $\{PID_{i,1}, PID_{i,2}, x_i, y_i, X_i, Y_i\}$ into L and returns (X_i, Y_i) to $\mathcal{A}_I$.

- Secret-Value-Extract-queries: After receiving the queries from $\mathcal{A}_I$ with $PID_i = (PID_{i,1}, PID_{i,2})$, $\mathcal{C}$ checks whether $\{PID_{i,1}, PID_{i,2}, x_i, y_i, X_i, Y_i\}$ exists in L. If it does, $\mathcal{C}$ returns x_i to $\mathcal{A}_I$. Otherwise, $\mathcal{C}$ does the Public-Key-Extract-queries to obtain (x_i, X_i, Y_i). After that, $\mathcal{C}$ adds $\{PID_{i,1}, PID_{i,2}, x_i, y_i, X_i, Y_i\}$ into L and returns x_i to $\mathcal{A}_I$.
- Public-Key-Replace-queries: After receiving the queries from $\mathcal{A}_I$ with $\{PID_{i,1}, PID_{i,2}, X_i', Y_i'\}$, $\mathcal{C}$ checks whether $\{PID_{i,1}, PID_{i,2}, x_i, y_i, X_i, Y_i\}$ exists in L. If it does, $\mathcal{C}$ sets $X_i = X_i'$, $Y_i = Y_i'$, $x_i = y_i = \perp$ and updates $\{x_i, y_i, X_i, Y_i\}$ into L. Otherwise, $\mathcal{C}$ sets $X_i = X_i'$, $Y_i = Y_i'$, $x_i = y_i = \perp$ and adds $\{PID_{i,1}, PID_{i,2}, x_i, y_i, X_i, Y_i\}$ to L.
- Sign queries: After receiving the queries from $\mathcal{A}_I$ with $\{PID_{i,1}, PID_{i,2}, M_i, T_i\}$, $\mathcal{C}$ retrieves the lists L, L_{H_1}, L_{H_3}, L_{H_4}, randomly selects $v_i \in Z_q^*$, $w_i \in Z_q^*$, $o_i \in Z_q^*$ and sets $s_i = v_i$, $t_i = w_i$, $e_i = o_i$, $\mathcal{K}_i = (x_1, y_1) = s_i \cdot G + t_i[Y_i + Z_i \cdot D_{pub} + \varphi_i \cdot X_i]$, $r_i = e_i + x_1 \text{mod} q$. $\mathcal{C}$ returns $\sigma_i = (r_i, s_i)$ to $\mathcal{A}_I$ and adds $H_1\{e_i, Z_i, M_i, T_i\}$ into L_{H_5}. For the output $\sigma_i = (r_i, s_i)$ of the signature oracle satisfies $\mathcal{K}'_{\,i} = (x'_{\,1}, y'_{\,1}) = s_i \cdot G + t_i[Y_i + Z_i \cdot D_{pub} + \varphi_i \cdot X_i]$, $R = e_i + x_1' \text{mod} q = r_i$.
- Forgery: After all queries have been completed, $\mathcal{A}_I$ outputs a forged tuple $\{M_i^*, PID_{i,1}^*, PID_{i,2}^*, T_i^*, \sigma_i^{*(1)}\}$. $\mathcal{C}$ verifies whether $\mathcal{K}_i{}^* = (x_1', y_1') = s_i^* \cdot G + t_i^*(Y_i + Z_i^* \cdot D_{pub} + \varphi_i^* \cdot X_i)$, $R^* = e_i^* + x_1^{\prime *} \text{mod} q = r_i^*$ holds. If it does not hold, $\mathcal{C}$ terminates the simulation. Otherwise, $\mathcal{C}$ replays the above process by choosing different H_1, H_3 and H_4 based on forking lemma. $\mathcal{A}_I$ will output three other distinct valid signatures $\sigma_i^{*(2)}$, $\sigma_i^{*(3)}$ and $\sigma_i^{*(4)}$.

Finally, we can obtain four equations as below.

$$k_i = s_i^{*(j)} + t_i^{*(j)}(g \cdot h_i + Z_i^{*(j)} \cdot K + \varphi_i^{*(j)} \cdot x_i), \text{ where } j = 1,2,3,4. \tag{10}$$

In the above four equations, k_i, g, K and x_i represent the discrete logarithms of $\mathcal{K}_i$, P_{pub}, D_{pub} and X_i, respectively, which are not known to $\mathcal{C}$. $\mathcal{C}$ can obtain the four unknown values by solving the above four linear independent equations, where g is the solution of ECDLP. □

Theorem 2. *CPPA-SM2 is existentially unforgeable under adaptive chosen-identity and chosen-message attacks against*$\mathcal{A}_{II}$*with the assumption that ECDLP is hard to resolve.*

Proof of Theorem 2. Let $\mathcal{C}$ be the solver of the ECDLP. Suppose that $\mathcal{A}_{II}$ can succeed in forging a valid signature by interacting with $\mathcal{C}$. $\mathcal{C}$ utilizes $\mathcal{A}_{II}$ to solve the ECDLP. Here, we give an ECDLP instance $\{G, G' = g \cdot G\}$. $\mathcal{C}$ executes the simulation to compute g through interacting with $\mathcal{A}_{II}$ as follows.

- Setup: On input $\{G, G'\}$, $\mathcal{C}$ sets $P_{pub} = s \cdot G$ and returns $\{p, q, s, E(F_p), G, Z_q^*, P_{pub}, H_1, H_2, H_3, H_4, H_5\}$ to $\mathcal{A}_{II}$. $\mathcal{A}_{II}$ selects $PID_{\,i}^* = (PID_{\,i,1}^*, PID_{\,i,2}^*)$ as a target vehicle. In addition, $\mathcal{C}$ maintains five lists $L = \{PID_{i,1}, PID_{i,2}, x_i, y_i, X_i, Y_i\}$, $L_{H_1} = \{h_i, X_i, P_{pub}\}$, $L_{H_3} = \{Z_i, len(PID_{i,2}), PID_{i,2}, a, b, G, X_i\}$, $L_{H_4} = \{\varphi_i, PID_{i,1}, T_i\}$, $L_{H_5} = \{e_i, Z_i, M_i, T_i\}$, which are empty initially.
- Query: $\mathcal{C}$ responds to -H_i-queries ($i = 1,3,4,5$), Partial-Private-Key-Extract-queries, Secret-Value-Extract-queries and Sign queries as in Theorem 1. $\mathcal{C}$ responds to Public-Key-Extract-queries as follows.
- Public-Key-Extract-queries: After receiving the queries from $\mathcal{A}_{II}$ with $PID_i = (PID_{i,1}, PID_{i,2})$, $\mathcal{C}$ checks whether $\{PID_{i,1}, PID_{i,2}, x_i, y_i, X_i, Y_i\}$ exists in L. If it does, $\mathcal{C}$ returns (X_i, Y_i) to $\mathcal{A}_{II}$. Otherwise, $\mathcal{C}$ does the Partial-Private-Key-Extract-queries to obtain y_i.
- If $PID_i = PID_i^*$, $\mathcal{C}$ sets $X_i = G' = g \cdot G$, $Y_i = y_i \cdot G$, $x_i = \perp$. $\mathcal{C}$ adds $\{PID_{i,1}, PID_{i,2}, x_i, y_i, X_i, Y_i\}$ into L and sends (X_i, Y_i) to $\mathcal{A}_{II}$.

- If $PID_i \neq PID_i^*$, $\mathcal{C}$ chooses $x \in Z_q^*$ randomly, computes $X_i = x \cdot G$, $x_i = x$, $Y_i = y_i \cdot G$. After that, $\mathcal{C}$ adds $\{PID_{i,1}, PID_{i,2}, x_i, y_i, X_i, Y_i\}$ into L and returns (X_i, Y_i) to $\mathcal{A}_{II}$.
- Forgery: After all queries have been completed, $\mathcal{A}_{II}$ outputs a forged tuple $\{M_i^*, PID_{i,1}^*, PID_{i,2}^*, T_i^*, \sigma_i^{*(1)}\}$. $\mathcal{C}$ verifies whether $\mathcal{K}_i{}^* = (x_1', y_1') = s_i^* \cdot G + t_i^*(Y_i + Z_i^* \cdot D_{pub} + \varphi_i^* \cdot X_i)$, $R^* = e_i^* + x_1'^{*} \bmod q = r_i^*$ holds. If it does not hold, $\mathcal{C}$ terminates the simulation. Otherwise, $\mathcal{C}$ replays the above process by choosing different H_3 and H_4 based on forking lemma. $\mathcal{A}_{II}$ will output two other distinct valid signatures $\sigma_i^{*(2)}$ and $\sigma_i^{*(3)}$.

  Finally, we can obtain three equations as below.

$$k_i = s_i^{*(j)} + t_i^{*(j)}(s \cdot h_i + Z_i^{*(j)} \cdot K + \varphi_i^{*(j)} \cdot x_i), \text{ where } j = 1,2,3. \tag{11}$$

In the above three equations, k_i, K and x_i represent the discrete logarithms of $\mathcal{K}_i$, D_{pub} and X_i, respectively, which are not known to $\mathcal{C}$. $\mathcal{C}$ can obtain the three unknown values by solving the above three linear independent equations, where x_i is the solution of ECDLP.

However, it is difficult to solve the ECDLP in polynomial time. So, under the random oracle model, CPPA-SM2 is existentially unforgeable under adaptive chosen-identity and chosen-message attacks. □

4.3. Informal Security Analysis

Anonymity and Privacy-Preserving: In the CPPA-SM2 scheme, vehicles use pseudonyms $PID_i = (PID_{i,1}, PID_{i,2})$ to communicate with other entities. To obtain the vehicle's real identity RID_i, the adversary must compute $RID_i = PID_{i,2} \oplus H(c_i \cdot P_{pub}) = PID_{i,2} \oplus H(c_i \cdot s \cdot G)$. However, due to the hardness of the Computational Diffie–Hellman (CDH) problem, the adversary is unable to obtain RID_i, thereby protecting the vehicle's identity privacy. Additionally, since vehicles participate in federated learning using pseudonyms, and these pseudonyms are updated with each message sent, even if external adversaries or RSUs gain access to the plaintext local model parameters, they cannot link them to specific vehicles. This prevents the inference of any private information, thus providing privacy protection during the federated learning process.

Traceability: When a vehicle with malicious behavior is detected, TA can trace its real identity $RID_i = PID_{i,2} \oplus H(s \cdot PID_{i,1})$ from its pseudonym $PID_i = (PID_{i,1}, PID_{i,2})$ using the system's master private key s.

Message integrity and authentication: According to Theorem 1 and Theorem 2, as long as the ECDLP is hard to solve, the CPPA-SM2 scheme is existentially unforgeable under adaptive chosen-identity and chosen-message attacks against the attackers $\mathcal{A}_I$ and $\mathcal{A}_{II}$.

Non-repudiation: Since only the message signer V_i can compute the signature key sgk_i, an adversary cannot forge valid signatures for a specific vehicle identity. Additionally, the TA can execute the Trace algorithm to obtain the vehicle's real identity. Therefore, once a vehicle's message passes the signature verification, it cannot be denied.

Un-linkability: Since the vehicle pseudonym identity PID_i is generated during the signing process and the random number used in the signature generation process is non-repetitive, each PID in every signature is unique. As a result, any adversary cannot link any number of signatures sent by the same vehicle.

Forward privacy: When a new vehicle joins the group C, the new group key K' is randomly generated by the TA and is independent of the old group key K. Therefore, the newly joined vehicle cannot access the group's communications prior to joining.

Backward privacy: When a vehicle is revoked or leaves the group, the TA will remove the legitimate information c_i associated with that vehicle from u and compute a new group key K' and group public key $\beta' = K' \cdot u$ and $D'_{pub} = K' \cdot G$. Since the revoked vehicle cannot obtain the updated group key K', it cannot access the communications after leaving the group.

Impersonation attack: If an adversary wants to impersonate vehicle V_i to the RSUs nearby or other vehicles V_j, they must generate a valid message $\{M_i, \sigma_i, (X_i, Y_i), PID_i, T_i\}$ that passes the verification algorithm. However, according to Theorem 1 and Theorem 2, it is evident that no polynomial adversary can forge a valid message.

Modification attack: According to Theorem 1 and Theorem 2, any modification of the message $\{M_i, \sigma_i, (X_i, Y_i), PID_i, T_i\}$ can be detected by the verification algorithm. Therefore, the proposed CPPA-SM2 scheme can withstand the modification attack.

Replay attack: In the proposed CPPA-SM2 scheme, vehicles use the current timestamp T_i when generating message signatures. Therefore, message verifiers can resist replay attacks by verifying the freshness of the timestamp T_i.

Collusion attack: Several vehicles would collaborate to try to compute the new group key K' after they left the group. However, since their legitimate information c_i has been removed from u, these leaving vehicles cannot conspire to calculate the new group key K'.

5. Performance Evaluation

In this section, we will evaluate the performance of the proposed CPPA-SM2 scheme from both security features, computation overhead and communication overhead perspectives, and compare and analyze it with the existing works. For bilinear pairings-based CPPA schemes for IoV, we construct a bilinear pairing $\overline{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_T$, where $\mathbb{G}_1$ is an additive group generated by a point $\overline{G}$ with the order $\overline{q}$ on the super singular elliptic curve $\overline{E} : y^2 = x^3 + x \bmod \overline{p}$ with embedding degree 2, $\overline{p}$ is a 512-bit prime number, $\overline{q}$ is a 160-bit prime number. For ECC-based CPPA schemes for IoV, we construct an additive group $\mathbb{G}$ generated by a point G with the order q on a non-singular elliptic curve $E : y^2 = x^3 + ax + b \bmod p$, where p, q are two 256-bit prime numbers and $a, b \in Z_p^*$. We calculate the execution time of basic cryptographic operations using the MIRACL library in VS 2019 with Windows 11 operating system over an Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz, as shown in Table 2.

Table 2. Execution time of basic cryptographic operations and element size.

Symbols	Meanings	Time (ms)/Size (Byte)
$T_{inverse}$	Time of module inverse on Z_q^*	0.0181 ms
T_{mod}	Time of mod operation on Z_q^*	0.0020 ms
T_e	Time of module exponential on Z_q^*	0.0434 ms
T_m	Time of module multiplication on Z_q^*	0.0044 ms
T_{SE}	Encryption time of AES algorithm	10.0761 ms
T_{DE}	Decryption time of AES algorithm	0.1759 ms
$T_{\oplus}$	Time of XOR operation	0.0009 ms
T_{bp}	Time of bilinear pairing	8.7985 ms
T_{bpm1}	Time of multiplication on bilinear group $\mathbb{G}_1$	0.1361 ms
T_{bpe1}	Time of exponential on bilinear group $\mathbb{G}_1$	1.3451 ms
T_{bpm2}	Time of multiplication on bilinear group $\mathbb{G}_2$	0.0069 ms
T_{bpe2}	Time of exponential on bilinear group $\mathbb{G}_2$	0.0869 ms
T_{em}	Time of scalar multiplication on ecliptic curve group $\mathbb{G}$	1.4944 ms
T_{ea}	Time of point addition on ecliptic curve group $\mathbb{G}$	0.1376 ms
T_h	Time of one-way hash function	0.3018 ms
T_{mtp}	Time of hash mapped to point	48.3228 ms
$\|T\|$	Size of timestamp	4 bytes
$\|ID\|$	Size of ID	8 bytes
$\|AES\|$	The ciphertext size of AES algorithm	32 bytes
$\|\mathbb{G}\|$	Size of elements on elliptic curve $\mathbb{G}$	64 bytes
$\|\mathbb{G}_1\|$	Size of elements on bilinear group $\mathbb{G}_1$	128 bytes
$\|\mathbb{G}_2\|$	Size of elements on bilinear group $\mathbb{G}_2$	128 bytes
$\left\|Z_q^*\right\|$	Size of elements on Z_q^*	32 bytes
$\|H\|$	Output size of hash function	32 bytes

5.1. Computation Costs

We compared the computational costs of the CPPA-SM2 scheme with other relevant schemes in terms of signature generation, single signature verification, batch verification and member management, as shown in Tables 3 and 4, and Figures 3 and 4, where "-" indicates that the property is not considered in the scheme, MS denotes the message sign and MV denotes the message verification.

Zhao et al. scheme [22] offers relatively low computational overhead, but RSU needs to send a request to TA for each identity verification, and there is a key escrow issue. In Kanchan et al. scheme [24] based on bilinear pairings, group signature is used instead of an individual signature for message authentication, and the group manager achieves tracing of malicious vehicles. Generating a group signature requires performing $T_h + 4T_{bp} + 4T_{bpe2} + 6T_e + 9T_m$. Verifying the group signature requires performing $T_h + 2T_m + 4T_{bpe2} + 5T_{bp} + 8T_e$, resulting in a relatively high computational overhead. In Jiang et al. scheme [26], similarly, bilinear pairing operations are used, requiring $2T_h + 5T_{bpe1}$ computations to generate a signature and $T_i + T_{bpe1} + T_{bpe2} + T_h + T_{DE} + T_{bpm2} + 3T_{bp}$ computations to verify the signature. In Yang et al. scheme [37], generating a signature requires performing $T_\oplus + 2T_h + 3T_{mtp} + 4T_{bpm1} + 6T_{bpe1}$. To verify the signature, $T_\oplus + T_{bpm1} + 2T_h + 3T_{bpe1} + 3T_{bpm2} + 5T_{bp} + 5T_{mtp}$ operations are needed. Due to the involvement of bilinear pairings and hash-to-point mappings, this method incurs the highest computational overhead. In Lin et al. scheme [38], a vehicle calculates $2T_h + 2T_{ea} + 3T_m + 6T_{em}$ to generate the anonymous public keys and a signature. Upon receiving the signature, RSU verifies it by performing $T_h + 3T_{em} + 4T_{ea}$. Additionally, Zhao et al. scheme [22], Kanchan et al. scheme [24], Jiang et al. scheme [26] and Lin et al. scheme all require maintaining a revocation list for revocation purposes, which incurs additional lookup and maintenance overhead. CPPA-SM2 does not require bilinear pairings or hash-to-point mappings, relying only on basic ECC operations, thus reducing computational costs. Specifically, when a vehicle sends a message, it first generates an unlinkable pseudonym PID_i by performing one T_{em}, one $T_\oplus$ and one T_h. Then, it generates the signature by performing three T_h, one T_{em}, four T_m and one T_i. Therefore, the computation cost for signature generation is $T_\oplus + T_i + 2T_{em} + 4T_m + 4T_h$. To authenticate the message sent by the vehicle, the RSU, upon receiving the message, needs to perform $3T_h + 3T_{ea} + 4T_{em}$. Therefore, the total computation cost for signature generation and signature verification in CPPA-SM2 is $T_\oplus + T_i + 3T_{ea} + 4T_m + 6T_{em} + 7T_h$. When RSU receives messages sent from n vehicles, it performs batch verification of the messages by executing $(2n+1)T_{ea} + (2n+2)T_{em} + 3nT_h$. To test the effectiveness of batch verification, we conducted experimental comparisons between CPPA-SM2 and Xiong et al. scheme [28] and Shen et al. scheme [39]. In batch verification, the RSU will verify the n messages received simultaneously from n vehicles, meaning n represents both the number of signatures received by the RSU at the same time and the number of vehicles. In the experiment, we tested with n set to 20, 40, 60 and 100, respectively. In CPPA-SM2, when RSU simultaneously receives n messages from n vehicles, it needs to compute three T_h, two T_{em} and two T_{ea} for each vehicle. Finally, it performs two T_{em} and one T_{ea} to verify multiple messages. Therefore, the total cost of batch verification is $3nT_h + (2n+2)T_{em} + (2n+1)T_{ea}$. In Xiong et al. scheme [28], it performs four T_h, two T_{em} and three T_{ea} for each vehicle. Then, it also executes three T_{em} and one T_{ea}. Therefore, the total cost of batch verification is $4nT_h + (2n+3)T_{em} + (3n+1)T_{ea}$. In Shen et al. scheme [39], RSU invokes one exponent operation, one bilinear pairing and one multiplication to confirm the equation $m = e(\eta, pk_i)e(P,P)^{-r_2}$. Its batch verification is based on $\prod_n e(\eta_n, pk_n)e(P,P)^{-r_{2,n}} = \prod_n m_n$, which needs n times T_{bp}, n times nT_{bpe2} and $(3n-2)T_{bpm1}$. The results are shown in Table 4 and Figure 4. From the experimental results, it can be seen that the batch-verification performance of our scheme is better than these two schemes. In terms of tracing cost, Kanchan et al. scheme [24], Yang et al. scheme [37], Lin et al. scheme [38] and CPPA-SM2 are 1.3451 ms, 0.1759 ms, 1.6320 ms and 0.3027 ms, respectively. All these approaches can achieve fast identity tracing. But in terms of revocation, all schemes except CPPA-SM2 utilize revocation lists, leading to additional maintenance and lookup overheads, while CPPA-SM2 only requires a single modular operation to efficiently revoke vehicles. Therefore, overall, compared to other schemes, CPPA-SM2 not only reduces the computational costs of signature generation and verification, and supports batch verification, but it also achieves efficient tracing and revocation of vehicles while preserving vehicle privacy.

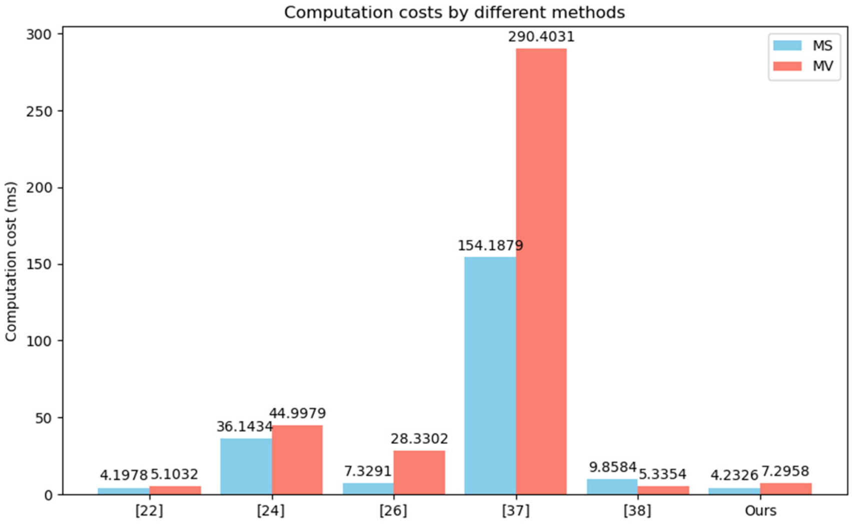

Figure 3. Comparison of computation costs.

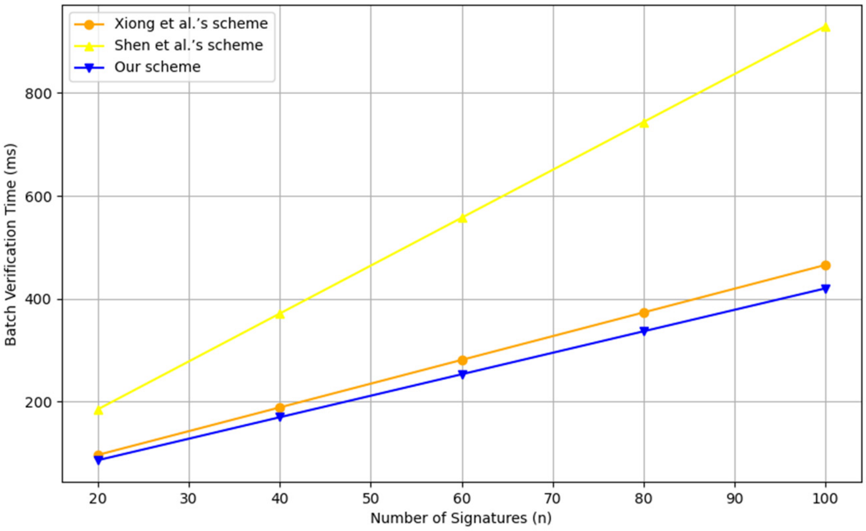

Figure 4. Comparison of the scheme proposed by [28,39], and our scheme in batch validation time.

Table 3. Analysis of computation costs for different schemes.

Scheme	MS	MV	Trace	Revoke
[22]	$2T_{\oplus} + 2T_{em} + 4T_h$	$2T_{\oplus} + 2T_{em} + 7T_h$	-	Revocation list
[24]	$T_h + 4T_{bp} + 4T_{bpe2} + 6T_e + 9T_m$	$T_h + 2T_m + 4T_{bpe2} + 5T_{bp} + 8T_e$	T_{bpe1}	Revocation list
[26]	$2T_h + 5T_{bpe1}$	$T_i + T_{bpe1} + T_{bpe2} + T_h + T_{DE} + T_{bpm2} + 3T_{bp}$	$O(1)$	Revocation list
[37]	$T_{\oplus} + 2T_h + 3T_{mtp} + 4T_{bpm1} + 6T_{bpe1}$	$T_{\oplus} + T_{bpm1} + 2T_h + 3T_{bpe1} + 3T_{bpm2} + 5T_{bp} + 5T_{mtp}$	T_{DE}	-
[38]	$2T_h + 2T_{ea} + 3T_m + 6T_{em}$	$T_h + 3T_{em} + 4T_{ea}$	$T_{em} + T_{ea}$	Revocation list
Ours	$T_{\oplus} + T_i + 2T_{em} + 4T_m + 4T_h$	$3T_h + 3T_{ea} + 4T_{em}$	$T_h + T_{\oplus}$	T_{mod}

Table 4. Comparison of batch-verification costs.

Scheme	Batch Verification Time
[28]	$4nT_h + (2n+3)T_{em} + (3n+1)T_{ea}$
[39]	$nT_{bp} + nT_{bpe2} + (3n-2)T_{bpm1}$
Ours	$3nT_h + (2n+2)T_{em} + (2n+1)T_{ea}$

5.2. Communication Costs

We compared the communication costs of CPPA-SM2 with other schemes, mainly including the following: the size of single signature (SSS), the total number of transmitted messages (NTMs), their sizes (STMs) and the number of interactions (NIs). The results are shown in Table 5 and Figure 5. In Zhao et al. scheme [22], to complete the authentication, interaction is required four times, making it the highest number of interactions. Its total computational cost is 476 bytes. The communication overhead for the group signature $\{D_1, D_2, D_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2}\}$ generated in Kanchan et al. scheme [24] is the highest, at 576 bytes. Jiang et al. scheme [26], Yang et al. scheme [37] and CPPA-SM2 all require only one interaction to complete message authentication. In Lin et al. scheme [38], vehicles need to transmit $\{\sigma_n, k_n, U_n, D_n, Z'_n\}$ for message authentication, with a total size of 480 bytes. In CPPA-SM2, the generated signature, denoted as $\sigma_i = (r_i, s_i)$, consists of two elements from Z_q^*; hence, its size is merely 64 bytes. To authenticate the signature, three additional messages $\{PID_i, (X_i, Y_i), T_i\}$ of size 228 bytes need to be transmitted, resulting in a total transmission cost of 292 bytes. In Yang et al. scheme [37], The generation of a single signature is denoted as $C_i = \{R_i, c_i, s_i\}$, where R_i, c_i and s_i belongs to $\mathbb{G}_1$; thus, the size of C_i is 384 bytes.

Table 5. Comparison of communication costs for different schemes.

Scheme	SSS	NTM	STM	NI
[22]	$\lvert ID\rvert + \lvert\mathbb{G}\rvert + \lvert T\rvert + 2\lvert Z_q^*\rvert$	4	$2\lvert ID\rvert + 2\lvert\mathbb{G}\rvert + 3\lvert T\rvert + 10\lvert Z_q^*\rvert$	4
[24]	$\lvert\mathbb{G}_2\rvert + 2\lvert\mathbb{G}_1\rvert + 6\lvert Z_q^*\rvert$	9	$\lvert\mathbb{G}_2\rvert + 2\lvert\mathbb{G}_1\rvert + 6\lvert Z_q^*\rvert$	2
[26]	$\lvert\mathbb{G}_1\rvert + \lvert Z_q^*\rvert$	5	$3\lvert\mathbb{G}_1\rvert + 3\lvert Z_q^*\rvert$	1
[37]	$3\lvert\mathbb{G}_1\rvert$	2	$3\lvert\mathbb{G}_1\rvert$	1
[38]	$2\lvert\mathbb{G}\rvert + 3\lvert Z_q^*\rvert$	4	$\lvert T\rvert + 3\lvert Z_q^*\rvert + 4\lvert\mathbb{G}\rvert$	2
Ours	$2\lvert Z_q^*\rvert$	4	$\lvert T\rvert + \lvert H\rvert + 2\lvert Z_q^*\rvert + 3\lvert\mathbb{G}\rvert$	1

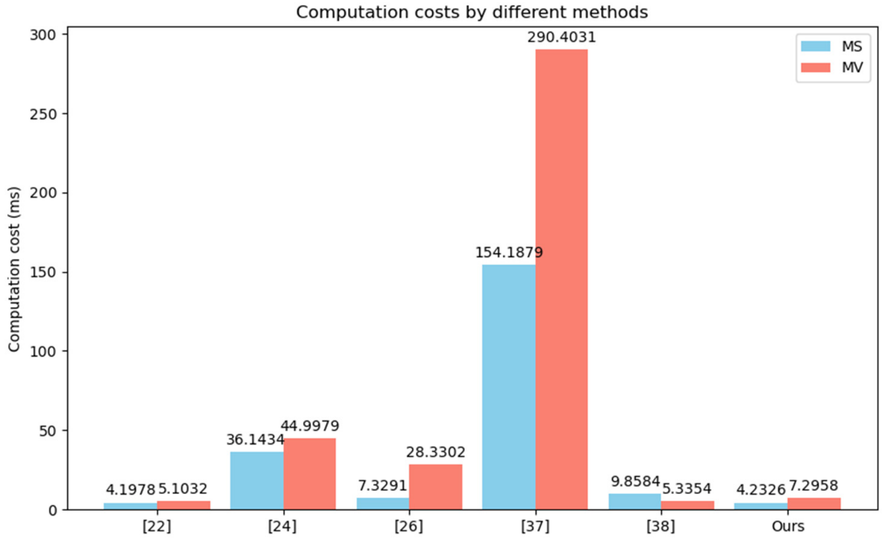

Figure 5. Comparison of communication costs.

In Lin et al. scheme [38], the obtained signature is denoted as $\{c_i, z_{i,1}, z_{i,2}, R_{i,1}, R_{i,2}\}$, with a length of 224 bytes. Additionally, to resist replay attacks, $\{ts_i, APK_a^1, APK_a^2\}$ are also sent, making the total message length for transmission 356 bytes. From the experimental results, it can be observed that CPPA-SM2 has the smallest signature size and total cost of transmitting messages. This makes it more suitable for operation in bandwidth-constrained vehicular networking environments.

5.3. Security Features

We compared the security features (SFs) satisfied by these schemes, including the following: 1: anonymity; 2: traceability; 3: authenticity; 4: integrity; 5: non-repudiation; 6: un-linkability; 7: forward security; 8: backward security; 9: key escrow-free; 10: batch verification; 11: revocability; 12: dynamic member management; and 13: un-forgeability. The results are shown in Table 6, where 1–13 represent these security features in order, with $\surd$ indicating that the security feature is met and × indicating that it is not met. From the results, it can be seen that all schemes achieve 1: anonymity, 3: authenticity, 4: integrity and 6: un-linkability. Zhao et al. scheme [22], Kanchan et al. scheme [24], Jiang et al. scheme [26] and CPPA-SM2 use digital signatures to verify the authenticity and integrity of the local model parameters uploaded by vehicles. However, in Zhao et al. scheme [22] and Kanchan et al. scheme [24], since TA possesses all users' private keys, there is a key escrow issue. Jiang et al. scheme [26] satisfies most of the security features; however, it uses a revocation list for identity management, resulting in additional verification and maintenance overhead. Furthermore, it does not support 12: dynamic member management. To achieve 6: un-linkability, Yang et al. scheme [37] and Lin et al. scheme [38] use a set of pseudonyms to hide real identities, whereas CPPA-SM2 achieves 6: un-linkability by randomly generating pseudonyms each time a signature is made. Overall, compared to these schemes, CPPA-SM2 achieves more comprehensive security attributes, supports 10: batch verification and 12: dynamic member management, and has lower computational and communication costs.

Table 6. Security features.

Scheme	SF												
	1	2	3	4	5	6	7	8	9	10	11	12	13
[22]	√	×	√	√	×	√	√	×	×	×	√	×	×
[24]	√	√	√	√	√	√	×	×	×	×	√	×	√
[26]	√	√	√	√	√	√	√	√	√	√	√	×	√
[37]	√	√	√	√	√	√	×	×	√	√	×	×	√
[38]	√	√	√	√	√	√	×	×	√	√	√	×	√
Ours	√	√	√	√	√	√	√	√	√	√	√	√	√

Overall, compared to the state-of-the-art scheme, Jiang et al. scheme [26], CPPA-SM2 reduces the cost of single signature generation and verification by 42.25% and 74.25%, respectively. In terms of communication overhead, CPPA-SM2 reduces it by 60% and 39.17%, respectively. While the performance of CPPA-SM2 in batch verification is not as good as Jiang et al. scheme [26], it supports dynamic member management, enabling efficient member addition and revocation, which results in increased batch-verification costs.

6. Conclusions

In this paper, we propose a conditional privacy-preserving identity-authentication protocol that provides privacy protection for vehicles participating in federated learning in the IoV. Unlike most existing privacy-preserving federated learning schemes, it does not require complex cryptographic operations or the introduction of random noise. Instead, it achieves privacy protection by using dynamic pseudonyms to obscure the connection between model parameters and the real identities of vehicles, thereby maintaining federated learning efficiency.

Moreover, CPPA-SM2 is a certificateless authentication scheme based on ECC, CRT and the SM2 digital signature algorithm. It enables efficient identity authentication and dynamic member management, and supports batch verification. Security proofs and analyses demonstrate that it can ensure the authenticity and integrity of local model parameters, achieving secure vehicle authentication. Experimental results show that, compared to existing advanced schemes, CPPA-SM2 offers high computational efficiency and low communication overhead. Additionally, its integration with standard algorithms endows it with the potential for widespread application.

However, the focus of this paper is on identity-authentication schemes and privacy protection in the federated learning process. There are still some malicious clients in the federated learning process that may launch data-poisoning attacks by uploading malicious local model parameters, thereby affecting the performance of the global model. Therefore, future research could integrate Byzantine robust detection schemes to achieve privacy-preserving Byzantine robust federated learning. Additionally, with the development of post-quantum algorithms, the ECDLP may be efficiently solved by post-quantum algorithms, making ECC-based authentication schemes no longer secure. Future work can explore quantum-resistant identity-authentication schemes, such as lattice-based cryptography.

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

Funding: This research was funded by the Ministry of Science and Technology of the People's Republic of China, the Research on Digital Identity Trust System for Massive Heterogeneous Terminals in Road Traffic System (Grant No. 2022YFB3104402).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

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

References

1. Duan, W.; Gu, J.; Wen, M.; Zhang, G.; Ji, Y.; Mumtaz, S. Emerging Technologies for 5G-IoV Networks: Applications, Trends and Opportunities. *IEEE Netw.* **2020**, *34*, 283–289. [CrossRef]
2. Elbir, A.M.; Soner, B.; Coleri, S.; Gunduz, D.; Bennis, M. Federated Learning in Vehicular Networks. In Proceedings of the 2022 IEEE International Mediterranean Conference on Communications and Networking (MeditCom), Athens, Greece, 5–8 September 2022; pp. 72–77. [CrossRef]
3. Khan, L.U.; Mustafa, E.; Shuja, J.; Rehman, F.; Bilal, K.; Han, Z.; Hong, C.S. Federated Learning for Digital Twin-Based Vehicular Networks: Architecture and Challenges. *IEEE Wirel. Commun.* **2024**, *31*, 156–162. [CrossRef]
4. Zhang, X.; Chang, Z.; Hu, T.; Chen, W.; Zhang, X.; Min, G. Vehicle Selection and Resource Allocation for Federated Learning-Assisted Vehicular Network. *IEEE Trans. Mob. Comput.* **2023**, *23*, 3817–3829. [CrossRef]
5. Cao, X.; Başar, T.; Diggavi, S.; Eldar, Y.C.; Letaief, K.B.; Poor, H.V.; Zhang, J. Communication-Efficient Distributed Learning: An Overview. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 851–873. [CrossRef]
6. Qu, Z.; Tang, Y.; Muhammad, G.; Tiwari, P. Privacy protection in intelligent vehicle networking: A novel federated learning algorithm based on information fusion. *Inf. Fusion* **2023**, *98*, 101824. [CrossRef]
7. Ni, R.; Lu, Y.; Yang, B.; Yang, C.; Liu, X. A federated pedestrian trajectory prediction model with data privacy protection. *Complex Intell. Syst.* **2024**, *10*, 1787–1799. [CrossRef]
8. XHu, X.; Li, R.; Wang, L.; Ning, Y.; Ota, K. A Data Sharing Scheme Based on Federated Learning in IoV. *IEEE Trans. Veh. Technol.* **2023**, *72*, 11644–11656. [CrossRef]
9. Sikarwar, H.; Das, D. A Novel MAC-Based Authentication Scheme (NoMAS) for Internet of Vehicles (IoV). *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 4904–4916. [CrossRef]
10. Wei, K.; Li, J.; Ding, M.; Ma, C.; Yang, H.H.; Farokhi, F.; Jin, S.; Quek, T.Q.S.; Poor, H.V. Federated Learning With Differential Privacy: Algorithms and Performance Analysis. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3454–3469. [CrossRef]
11. Zhao, Y.; Zhao, J.; Yang, M.; Wang, T.; Wang, N.; Lyu, L.; Niyato, D.; Lam, K.-Y. Local Differential Privacy-Based Federated Learning for Internet of Things. *IEEE Internet Things J.* **2021**, *8*, 8836–8853. [CrossRef]
12. Zhou, H.; Yang, G.; Dai, H.; Liu, G. PFLF: Privacy-Preserving Federated Learning Framework for Edge Computing. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 1905–1918. [CrossRef]
13. Zhou, C.; Fu, A.; Yu, S.; Yang, W.; Wang, H.; Zhang, Y. Privacy-Preserving Federated Learning in Fog Computing. *IEEE Internet Things J.* **2020**, *7*, 10782–10793. [CrossRef]

14. Ma, Z.; Ma, J.; Miao, Y.; Li, Y.; Deng, R.H. ShieldFL: Mitigating Model Poisoning Attacks in Privacy-Preserving Federated Learning. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 1639–1654. [CrossRef]
15. Hijazi, N.M.; Aloqaily, M.; Guizani, M.; Ouni, B.; Karray, F. Secure Federated Learning with Fully Homomorphic Encryption for IoT Communications. *IEEE Internet Things J.* **2024**, *11*, 4289–4300. [CrossRef]
16. ZZhang, Z.; Wu, L.; Ma, C.; Li, J.; Wang, J.; Wang, Q.; Yu, S. LSFL: A Lightweight and Secure Federated Learning Scheme for Edge Computing. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 365–379. [CrossRef]
17. Taheri, R.; Shojafar, M.; Alazab, M.; Tafazolli, R. Fed-IIoT: A Robust Federated Malware Detection Architecture in Industrial IoT. *IEEE Trans. Ind. Inform.* **2021**, *17*, 8442–8452. [CrossRef]
18. Taheri, R.; Arabikhan, F.; Gegov, A.; Akbari, N. Robust Aggregation Function in Federated Learning. In *Advances in Information Systems, Artificial Intelligence and Knowledge Management*; Saad, I., Rosenthal-Sabroux, C., Gargouri, F., Chakhar, S., Williams, N., Haig, E., Eds.; ICIKS 2023. Lecture Notes in Business Information Processing; Springer: Cham, Switzerland, 2024; Volume 486. [CrossRef]
19. Al Sibahee, M.A.; Nyangaresi, V.O.; Abduljabbar, Z.A.; Luo, C.; Zhang, J.; Ma, J. Two-Factor Privacy-Preserving Protocol for Efficient Authentication in Internet of Vehicles Networks. *IEEE Internet Things J.* **2024**, *11*, 14253–14266. [CrossRef]
20. Ou, Z.; Xing, X.; He, S.; Wang, G. TDS-NA: Blockchain-based trusted data sharing scheme with PKI authentication. *Comput. Commun.* **2024**, *218*, 240–252. [CrossRef]
21. Chen, Y.; Su, Y.; Zhang, M.; Chai, H.; Wei, Y.; Yu, S. FedTor: An Anonymous Framework of Federated Learning in Internet of Things. *IEEE Internet Things J.* **2022**, *9*, 18620–18631. [CrossRef]
22. Zhao, P.; Huang, Y.; Gao, J.; Xing, L.; Wu, H.; Ma, H. Federated Learning-Based Collaborative Authentication Protocol for Shared Data in Social IoV. *IEEE Sens. J.* **2022**, *22*, 7385–7398. [CrossRef]
23. Zhang, J.; Cui, J.; Zhong, H.; Chen, Z.; Liu, L. PA-CRT: Chinese Remainder Theorem Based Conditional Privacy-Preserving Authentication Scheme in Vehicular Ad-Hoc Networks. *IEEE Trans. Dependable Secur. Comput.* **2019**, *18*, 722–735. [CrossRef]
24. Kanchan, S.; Choi, B.J. An Efficient and Privacy-Preserving Federated Learning Scheme for Flying Ad Hoc Networks. In Proceedings of the ICC 2022—IEEE International Conference on Communications, Seoul, Republic of Korea, 16–20 May 2022; pp. 1–6. [CrossRef]
25. Lin, H.-T.; Jhuang, W.-L. Blockchain-Based Lightweight Certificateless Authenticated Key Agreement Protocol for V2V Communications in IoV. *IEEE Internet Things J.* **2022**, 15. [CrossRef]
26. Jiang, Y.; Zhang, K.; Qian, Y.; Zhou, L. Anonymous and Efficient Authentication Scheme for Privacy-Preserving Distributed Learning. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 2227–2240. [CrossRef]
27. Ma, Y.; Cheng, Q.; Luo, X. 2PCLA: Provable Secure and Privacy Preserving Enhanced Certificateless Authentication Scheme for Distributed Learning. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 5876–5889. [CrossRef]
28. Xiong, H.; Chen, J.; Mei, Q.; Zhao, Y. Conditional Privacy-Preserving Authentication Protocol With Dynamic Membership Updating for VANETs. *IEEE Trans. Dependable Secur. Comput.* **2020**, *19*, 2089–2104. [CrossRef]
29. Zhong, H.; Wang, L.; Cui, J.; Zhang, J.; Bolodurina, I. Secure Edge Computing-Assisted Video Reporting Service in 5G-Enabled Vehicular Networks. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 3774–3786. [CrossRef]
30. Yuan, X.; Liu, J.; Wang, B.; Wang, W.; Li, T.; Ma, X.; Pedrycz, W. FedComm: A Privacy-Enhanced and Efficient Authentication Protocol for Federated Learning in Vehicular Ad-Hoc Networks. *IEEE Trans. Inf. Forensics Secur.* **2023**, *19*, 777–792. [CrossRef]
31. Zhang, Y.; Lei, H.; Wang, B.; Wang, Q.; Lu, N.; Shi, W.; Chen, B.; Yue, Q. Traceable ring signature schemes based on SM2 digital signature algorithm and its applications in the data sharing scheme. *Front. Comput. Sci.* **2024**, *18*, 182815. [CrossRef]
32. *GM/T 0003.2-2012*; SM2 Elliptic Curve Public Key Cryptographic Algorithm Part 2: Digital Signature Algorithm. National Standard of the People's Republic of China: Beijing, China, 2012.
33. Eltaras, T.; Sabry, F.; Labda, W.; Alzoubi, K.; Ahmedeltaras, Q. Efficient Verifiable Protocol for Privacy-Preserving Aggregation in Federated Learning. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 2977–2990. [CrossRef]
34. Maurya, C.; Chaurasiya, V.K. Efficient Anonymous Batch Authentication Scheme with Conditional Privacy in the Internet of Vehicles (IoV) Applications. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 9670–9683. [CrossRef]
35. Horng, S.-J.; Tzeng, S.-F.; Pan, Y.; Fan, P.; Wang, X.; Li, T.; Khan, M.K. b-SPECS+: Batch Verification for Secure Pseudonymous Authentication in VANET. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 1860–1875. [CrossRef]
36. Cui, J.; Zhang, J.; Zhong, H.; Xu, Y. SPACF: A Secure Privacy-Preserving Authentication Scheme for VANET With Cuckoo Filter. *IEEE Trans. Veh. Technol.* **2017**, *66*, 10283–10295. [CrossRef]
37. 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.* **2021**, *17*, 317–331. [CrossRef]
38. Lin, C.; Huang, X.; He, D. EBCPA: Efficient Blockchain-based Conditional Privacy-preserving Authentication for VANETs. *IEEE Trans. Dependable Secur. Comput.* **2022**, *20*, 1818–1832. [CrossRef]
39. Shen, J.; Liu, D.; Chen, X.; Li, J.; Kumar, N.; Vijayakumar, P. Secure Real-Time Traffic Data Aggregation with Batch Verification for Vehicular Cloud in VANETs. *IEEE Trans. Veh. Technol.* **2019**, *69*, 807–817. [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.