A Privacy-preserving Aggregation Scheme with Continuous Authentication for Federated Learning in VANETs

Xia Feng, Xiaofeng Wang, Haiyang Liu, Haowei Yang, Liangmin Wang.

***Abstract*—Federated Learning (FL) allows the collaborative training of a global model in Vehicular Ad-hoc Networks (VANETs): data is maintained on the owner's device and the local gradient updates to the model are aggregated through a secure protocol. However, despite its many advantages, FL is vulnerable to various attacks from malicious clients. Current defenses have several weaknesses. For example, the server may still select malicious clients for aggregation even after they have been identified in previous rounds. They haven't considered the continual monitoring of clients to resist their possible defection or collusion that occurs throughout the FL training process. In response to the weaknesses, we describe a new aggregation model with continuous authentication suits for VANETs. The authentication implementation relies on a non-interactive zero-knowledge proof which preserves privacy. We also minimize the computation and communication overhead by designing a two phases aggregation scheme, while introducing the Edge Devices (EDs) to assist the FL procedure. Finally, we introduce an application of such a model for VANETs. We describe the prototype implementation and experimentally confirm that the aggregation overhead of the client grows linearly and achieves training speed up over the prior work.**

***Index Terms*—Aggregation, Privacy-preserving, Authentication, VANETs.**

I. Introduction

DURING the last decade, we have witnessed the emergence of autonomous vehicles and a large number of sensors have been deployed with the intention of predominantly facilitating autonomous driving. Sensors used in vehicles including the camera, radar, Linda, and sonar enable the vehicles to capture the surrounding environment accurately, which is the basis for autonomous driving [1]. The widespread expansion of autonomous vehicles and driving technologies require the utilization of Machine Learning (ML) to derive precise decisions from the raw data produced by the sensors. However, besides using these sensor data locally, mining these onboard data with the surrounding sensors would obtain more stable, low-latency, and high-reliability services. Thus, there is a trend to cooperatively train a global model through Ad-hoc Vehicular Networks (VANETs), which is Federated Learning (FL), a distributed deep learning paradigm [2]. FL constructed over VANETs attracts attention because it enables ML over large-scale devices for developing innovative tasks, with data localized on end. Technically, each client trains a shared global model with a local dataset and uploads the gradient parameters to the server (aggregator) for aggregation. Afterward, the server shares the aggregation results with each client. Repeating this process until the training output converges.

As with all new technologies, there are risks. Although the FL training process is considered to be given the levels of privacy and security required by the distributed framework, a host of studies have still shown that various attacks are active in the gradient aggregation process.

It is possible for servers to analyze local updates for malicious usage that invade clients' privacy. Meanwhile, malicious clients may modify data features or provide a poisoned update that can take control of the entire training process, thereby rendering the final model invalid [3] [4]. Particularly, attackers can still indirectly obtain sensitive information including labels [5] [6], and memberships [7] [8] based on the shared gradients. Therefore, in heterogeneous environments such as the vehicular network, secure aggregation protocols [9]–[13] have been proposed to address this problem by applying cryptographic techniques, which suffices for the security and confidentiality of uploaded local parameters. For example, verifiable FL has been proposed [14] to enhance the system security and help build trust among legitimate participants. In its architecture, the FL clients need verification from the server in order to participate, including verification for clients' local information, including but not limited to identity and local data. However, these approaches haven't considered the continual monitoring of clients to resist their possible defection or collusion that occurs throughout the FL training process. More importantly, to ensure security, the access of FL clients should not be granted statically but continuously re-evaluated.

In this work, we introduce a novel privacy-preserving aggregation model for FL, which is scalable, efficient, and lightweight enough to be effective in a vehicular network setting. Our main contributions to this paper are described as follows:

- **Privacy-preserving aggregation scheme.** Our protocol introduces an FL aggregation architecture of continual authentication and evaluates ongoing training procedures. It develops a series of mechanisms to choose verifiable

Xia Feng is with the School of Automotive and Transportation Engineering, Jiangsu University, PRC. (e-mail: xiazio@ujs.edu.cn).

Xiaofeng Wang is with the School of Automotive and Transportation Engineering, Jiangsu University, PRC.(email:2212104045@stmail.ujs.edu.cn).

Haiyang Liu is with the School of Computer Science and Communication Engineering, Jiangsu University, PRC. (e-mail: haiyang@stmail.ujs.edu.cn).

Haowei Yang is with the School of Automotive and Transportation Engineering, Jiangsu University, PRC. (e-mail: 1065570220@qq.com).

Liangmin Wang is with School of Cyber Science and Engineering, Southeast University, PRC. (email: liangmin@seu.edu.cn).

(Corresponding author: Xia Feng.)

clients and block misbehaving clients and adversaries. To the best of our knowledge, this is the first work to investigate the FL aggregation scheme in VANETs that strives to achieve a balance of security, cost-efficiency, and privacy-preserving.

- **Efficiency and concurrency.** Our proposed scheme achieves efficiency, which benefits from two aspects. The first is the design of two separate phases, with each phase defined by a single communication round. The setup phase is executed only once, while the online phase is performed for every round of FL, which reduces the communication and computation overhead. The second is the low-latency authentication scheme that checks the message for validity based on a non-interactive zero-knowledge proof protocol without introducing additional communication rounds.
- **Applications for the vehicular network.** We implement our scheme into a prototype and is available as open source at https://github.com/xiazio1983/privacyFL. We experimentally confirm that the training overhead per client grows linearly and achieves training speed up over the prior work. We also implement the protocol using transportation-related datasets, demonstrating the practical performance with high accuracy and coverage rate.

The remainder of our paper is organized as follows: We survey existing protocols and background in Section II. The building blocks and some preliminaries are introduced in Section III. The problem statement is formalized in Section IV, as well as the introduction of the hierarchical architecture of our scheme. We describe the protocol in Section V, including the algorithms. The security and privacy of the model are analyzed in Section VI. We evaluate the performance and compare it with the prior work in Section VII. Finally, we conclude our proposed scheme and present the future work in Section VIII.

II. Background and Related Work

In this section, we introduce the background of FL in VANETs and survey existing techniques to present the requirements for our scheme. We also highlight the stringent privacy requirements.

A. FL in VANETs

FL enables distributed machine learning that accomplishes large-scale data mining tasks aiming at the privacy protection of clients. In vehicular network scenarios, sensors equipped on vehicles generate a large amount of data containing abundant client privacy information, e.g., location information, user preference, and driving trajectory [15]. With the help of powerful FL methods, these data are analyzed and utilized to achieve intelligent transportation systems [16]. Specifically, FL [17] in VANETs is a concept for distributed machine learning that links n clients (e.g. vehicles, mobiles) and several servers who collaboratively build a global model ψ. For VANETs, Road-side Units (RSUs) or Edge Devices (EDs) could act as servers. In each training round, RSU chooses a subset of the k clients and sends the current global model ψ to clients. Clients train ψ_t on their local data for multiple epochs. Afterward, each client sends the local gradient updates G_{t+1} to the server. Once the server has gathered enough updates, it aggregates the local gradients into the global model ψ_{t+1}. The process will be repeated until the FL training output converges.

B. FL Challenges in VANETs

Research focus on FL applied in VANETs faces two severe challenges: privacy protection and data quality [18]–[20]. In VANETs, clients and servers are communicated via open wireless channels in most cases, adversaries may launch various attacks. an attacker, e.g., a malicious central server, can easily reconstruct the participant's data with pixel-wise accuracy for images and token-wise matching for texts. To mitigate such so-called "deep leakage from gradients", privacy-awareness aggregation protocols are leveraged to protect the gradients uploaded by participants. Studies focus on aggregation schemes [21]–[23] tried to preserve clients' privacy while achieving great performance. Lu et al. [21] proposed DPAFL, which built a secure and robust FL scheme, using differential privacy for local model updates in VANETs. Yu et al. [22] designed a blockchain-based aggregation protocol, ensuring the stakes of legitimate clients and the accuracy of the system model.

For the aspect of data quality, vehicles acting as clients have limited communication bandwidth and compute power. Moreover, dynamic networking environments indirectly result in some unintentional behaviors of clients. The clients may update low-quality models caused by high-speed mobility or energy limitation. To address the challenges, Samarakoon et al. [24] proposed a FL approach for joint power and resource allocation, considering the vehicle-to-vehicle (V2V) communications delays incurred over wireless links. Further, Chen et al. [25] presented a probabilistic devices selection scheme, minimizing the FL convergence time as well as enhancing the FL performance. Hammoud et al. [16] leveraged fog computing devices to form stable fog federations to expand the geographical footprints. In their architecture, fog devices belonging to the same federations could optimize services in an adaptive fashion for FL tasks. Hu et al. in [26] presented a double-layer FL structure by coordinating multiple RSUs to improve learning performance. Hossain et al. [27] designed an FL-based vehicle cooperative positioning system to achieve high-precision coordinated vehicle positioning while ensuring client privacy. In their works, the two factors identified above are expressed as an optimization problem, which aims to maximize the performance of FL applications. Kang et al. [23] utilized blockchain to build an aggregation protocol, using reputation to measure the reliability the mobile devices. By encouraging high-quality workers to participate in FL, their scheme achieved secure and higher performance.

Based on the aforementioned discussion, most research efforts focused on optimizing the learning process in FL failed to address the tradeoffs between data quality, communication latency, and privacy protection. Specifically, 1) approaches that aim at providing private aggregation schemes often introduce additional operations; 2) approaches using an adaptive client

select strategy may still select malicious clients for aggregation even after they have been identified in previous rounds.

III. Preliminary

We would like to offer a brief technical preliminary knowledge to explain our protocol.

A. Bilinear Pairing

- $\mathbb{G}_1$ and $\mathbb{G}_2$ are two additive cyclic groups of prime order $q = \Theta(2^k)$. Let $\mathbb{G}_T$ denote a multiplicative group of the same prime order q.
- Let g_1 be a generator of $\mathbb{G}_1$ and g_2 a generator of $\mathbb{G}_2$.
- e: Let $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ be a bilinear mapping with the following properties:
  - Bilinear: For all $U, V \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p^*$, $e(aU, bV) = e(U, V)^{ab}$.
  - Non-degeneracy: $e(U, V) \neq 1_{\mathbb{G}_T}$.
  - Computability: There exists an efficient algorithm for computing $e(U, V)$.

B. Mathematical Assumptions

We state the following mathematical assumptions as the basis for our scheme.

- **q-Strong Diffie-Hellman Problem** [28] The q-SDH problem in additive group $\mathbb{G}_1, \mathbb{G}_2$ (where possibly $\mathbb{G}_1 = \mathbb{G}_2$) is defined as follows: given the $(q+2)$ tuple (g_1, g_2, γg_2, $\gamma^2 g_2$, ... , $\gamma^q g_2$), it is hard to fine x and calculate $(x, \frac{1}{x+\gamma} g_1)$.
- **Elliptic Curve Diffie-Hellman Problem** [29] Setting an additive cyclic group $\mathbb{G}_1$ constructed with an elliptic curve E, and the generator of $\mathbb{G}_1$ is g_1, the order is q. The ECDHP problem is defined as follows: given elements in $\mathbb{G}_1$: V, R, where $V = vg$, $R = rg$, $v, r \in \mathbb{Z}_p^*$, it is hard to compute vrg with knowledge V, R, g.
- **Elliptic Curve Discrete Logarithm Problem** [30] Setting an additive cyclic group $\mathbb{G}_1$ constructed with an elliptic curve E, and the generator of $\mathbb{G}_1$ is g_1, the order is q. The ECDHP problem is defined as follows: given the elements in $\mathbb{G}_1$: R, where $R = rg$, $x \in \mathbb{Z}_p^*$, it is intractable to compute r with knowledge R, g in probability polynomial time.

C. Pseudorandom Generator

A secure Pseudorandom Generator (**PRG**) can be defined as a generator that takes a random seed as input and produces a random number as output. The output space of the **PRG** can be denoted as $\mathbb{Z}_p^*$. From the security perspective, **PRG** guarantees that its output on random seed is computationally indistinguishable, as long as the seed is hidden from the distinguisher.

D. Authentication Mechanism

The authentication operation mainly contains key generation, signature generation, and verification, which involves four functions: **KG.param**, **KG.gen**, **SIG.sign** and **SIG.ver**.

- **KG.param** $(k) \rightarrow pp$: The algorithm produces public parameters pp based on the security parameter k, such as elliptic curve parameters. Particularly, based on the selected elliptic curve, the algorithm further constructs an additive cyclic group $\mathbb{G}_1$ with generator g_1 and order q. With the random elements h, μ, and ν from $\mathbb{G}_1$, TA selects two random numbers s_1 and s_2 from the set Z_q^* as the master secret keys, satisfying the equation $s_1\mu = s_2\nu = h$. TA also sets a random number γ from Z_q^* and computes $\Gamma = \gamma g$ as the group public key.
- **KG.gen** $(pp) \rightarrow (sk_i, pk_i, (A_i))$: This algorithm generates a private-public key pair for client i by elliptic curve cryptography (ECC). Each client chooses $x_i \in Z_q^*$ as membership secret key sk_i, and computes public key $pk_i = x_i \cdot g$. The element A_i is a group public key, which is used for the authentication process. Using γ, **KG.gen** computes $A_i = \frac{1}{x_i+\gamma} g$. Specifically, (A_i, x_i) is an SDH tuple.
- **SIG.sign** $(sk_i, A_i, M) \rightarrow \sigma$: The algorithm represents generation operation of the signature. Given sk_i, A_i, and message $M \in \{0,1\}^*$, generate the signature σ.
- **SIG.ver** $(\sigma, M) \rightarrow T/F$: The algorithm represents the verification operation of signature σ. Based on the zero-knowledge proof, checking whether the signature generator has a valid sk_i, A_i, and the integrity of signature message M.

Authenticated encryption guarantees confidentiality and integrity for messages exchanged between two or more parties. For our scheme, an authentication scheme is indispensable to ensure messages are sent by legitimate entities and not altered during transmissions [31]. Furthermore, the Trust Authority (TA) traces the original identities of malicious vehicles in a case of misconduct.

E. Key Agreement

The key agreement process is executed by the function **KA.agree**.

- **KA.agree** $(sk_i, pk_j) \rightarrow s_{i,j}$: The algorithm supports any client i to compute a private session key $s_{i,j}$ by combining its private session key sk_i with the public key pk_j of others (generated using the same pp). The specific Key Agreement scheme we will use is Diffie-Hellman key agreement [32], composed with a hash function. More specifically, $KG.param(k) \rightarrow pp$ samples group $\mathbb{G}_1$ with generator g_1 and order q, and a hash function H. **KA.gen**$(pp) \rightarrow (x_i, x_i \cdot g)$ samples a random $x_i \in Z_q^*$ as the secret key sk_i, and $x_i \cdot g$ as the public key pk_i, and **KA.agree**$(x_i, x_j \cdot g) \rightarrow s_{i,j}$ outputs $s_{i,j} = H((x_j \cdot g) \cdot x_i)$.

IV. Architecture of FL in VANETs

FL in VANETs should consider the dynamic behavior of vehicles, which may cause a degradation in the service

quality and deviations from the privacy-preserving aggregation protocol. Therefore, a solution must be devised to address the privacy concern as well as maintain a satisfactory quality to cope with the dynamic environment. Due to its unique components, FL in VANETs necessitates a specialized architecture that differs from other applications. In this section, we introduce an architecture designed specifically to facilitate FL using the VANETs concept.

Fig.1 summarizes the proposed architecture. TA is fully trusted and divides the whole precinct into several domains. In Domain A, the architecture composes a set of vehicles V_i equipped with On-board Units (OBUs), several EDs E_j, and one RSU R distributing uniformly. Vehicles located in Domain A are requesting the service from corresponding E_j and R. Nevertheless, vehicles in Domain D are being served by the nearby EDs due to wireless network limitations. We describe the workflow and the entity below.

A. Workflow

FL in VANETs is a concept for distributed machine learning that links n clients (e.g. vehicles, mobiles) and several servers (RSUs, EDs) that collaboratively build a global model ψ. In each training round, ED chooses a subset of the K clients and sends the current global model ψ^t to them. Clients train ψ^t on its local data for multiple epochs. Afterward, each client sends the local gradient updates G^{t+1} to ED. Once ED has gathered enough updates, it verifies the local gradients and signs them before sending the local gradients to RSU. After re-authentication, RSU identifies malicious clients based on their gradients. Thus, RSU could rationally select participants that are more likely to be honest and useful. RSU then sums up the local gradients from the selected clients into the global model ψ^{t+1}. During the process, the malicious clients will be added to the revocation list so that the adversary couldn't join in the next round of FL. The process will be repeated until the ML training output converges.

B. Entities

TA: TA serves as a reliable management center, duly authorized by legal regulations. Its primary responsibility lies in generating system parameters for vehicles, RSUs, and EDs. Within our system, the TA assumes the exclusive role of auditing the network, possessing the capability to disclose the true identities of any vehicles, RSUs, or EDs engaged in misconduct.

EDs: EDs are independent entities that provide computing and networking resources for the model. Each ED is responsible for 1) validating the vehicles and conducting key agreement protocol with vehicles; 2) collecting the masked local gradients; and 3) forwarding the masked local updates to RSU.

RSUs: RSUs are located alongside the roads to organize and coordinate vehicular communications in an optimized manner. They are authorized by TA and responsible for 1) validating the EDs, 2) identifying malicious clients, 3) evaluating the client's trust level, 4) aggregating the local gradients, and 5) broadcasting the global model to the vehicles.

Vehicles: In our model, vehicles act as clients of FL, responsible for training the model and uploading the local gradients. They rely on public wireless communication channels to communicate with other entities in VANETs.

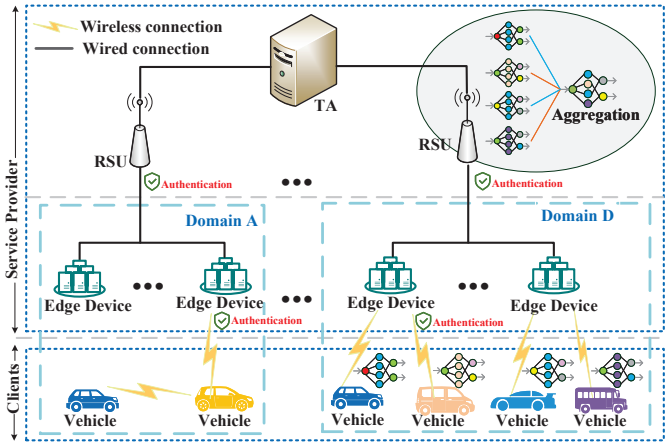


Fig. 1. The system model.

C. Characterize Threat Model

We assume that all entities in our model agree upon a model aggregation orchestrated by the TA. They all have a common interest in soundness (i.e., updating the correct local model, aggregating the global model) and preserving privacy (i.e., hiding the local model updates from each other and the server) [13]. Intuitively, our aggregation scheme performs between the vehicles, RSUs, and EDs, which communicate via open wireless channels in most cases. We define the threat model by characterizing the potential activities of entities. TA and RSU in our system are trust authorities. We considered two attack models $\mathcal{A}_1$ and $\mathcal{A}_2$ on the basis of EDs in the system model.

1) Semi-honest Construction: $\mathcal{A}_1$ is under the "semi-honest" assumption. EDs in the system model are under the "honest-but-curious" assumption. They may be curious about the client's private information but would never conspire with the adversaries. In $\mathcal{A}_1$, EDs may analyze the received local gradient updates and try to infer the data content or distribution. Meanwhile, the malicious clients can run any arbitrary polynomial-time attack strategy where these attacks may impact gradient aggregation and the quality of service (QoS) of FL.

2) Semi-malicious Construction: $\mathcal{A}_2$ is under the "semi-malicious" assumption. Our setting is analogous to the assumption in Bell. et al [33], extending the ideas behind its protocol to withstand an adversary that controls an ED and a fraction of clients. Specifically, any ED may collude with clients in order to invade other clients' data.

D. Design Goal

We then present our design goal with semi-honest construction and semi-malicious construction, under the consideration of correctness and integrity, and privacy-preserving.

- **Correctness and integrity:** On behave of the correctness property, the authorized clients can always be verified that they are legal entities. On behave of the integrity property, the gradients sent by the authorized clients can always be proved correct without being modified or forged.
- **Privacy-preserving:** Our scheme should have the ability to update the local gradients to aggregators without revealing the identity of the clients. The RSUs or EDs cannot sufficiently distinguish whether two or more messages are related or from the particular client.

V. FL Aggregation Architecture

In this section, we describe the FL architecture in VANETs, which ensures the clients who join the FL are continuously authenticated, as shown in Fig. 2 and 3.

A. FL Aggregation Procedure

We design the aggregation protocol by inspecting and verifying every single access. The protocol is composed of a **Registration Phase** and an **Online Phase**, with each phase defined by a single communication round. As shown in Fig. 2, the **Registration Phase** is executed only once, while in Fig.3, the **Online Phase** is performed for every round of FL. Our construction works on FL with security and privacy enhancement, enabling significant efficiency improvements. The frequently used notations are listed in Table I.

B. The Registration Phase

During the registration process, the TA is responsible for generating the public parameters for the system. Firstly, TA defines an elliptic curve E. Based on the point of E and the infinity point Θ, TA constructs an additive group $\mathbb{G}_1$ of order q with the generator g. Next, TA selects a security parameter k and generates the parameters pp using the function **KG.param**(k). Finally, TA sets the public parameters $pp = (E, \mathbb{G}_1, g, q, h, \mu, \nu, \Gamma)$ and distributes pp to the entities in the system.

The client i registers by generating the group-public key $(sk_i^1, pk_i^1, A_i) \leftarrow$ **Key.Gen**(pp) and session key pair (sk_i^2, pk_i^2). To assist the FL procedure, EDs, and RSUs register to the system using the public parameters as well. The specifications of the setup phase are given in Fig. 2.

C. The Online-Masking Phase

In this phase, each client conducts local training with its dataset and masks the gradients. After that, the client signs the masked gradients and forwards them to ED which is within its communication converge. The specifications of this phase are given in Fig. 3.

Specifically, we explain the steps as follows. The clients first take the initial global model ψ, and the number of training rounds TR as input. Then, in each round $tr \in [2, TR]$, a random subset of clients $U \subset K$, is chosen following the design of FL to train the local model. In the masking step, the client generates a random seed which is further extended to mask the local gradients. More concretely, client i executes a key agreement protocol **KA.agree** with RSU and generates the session key $s_{r,i}$ as below:

$$s_{r,i} \leftarrow \textbf{KA.agree}(sk_r^2, pk_i^2). \tag{1}$$

Then the session key is leveraged as a **PRG** input and produces a random matrix to mask the local gradients:

$$R_{i,r} = \textbf{PRG}(s_{i,r}). \tag{2}$$

Similarly, the RSU generates $s_{r,i} \leftarrow KA.agree\left(sk_i^2, pk_r^2\right)$, which is used to unmask the gradients.

After that, client i masks the local gradient matrix as Equation (3).

$$Msk\{G_i^{t+1}\} = R_{i,r} + G_i^{t+1}. \tag{3}$$

Then, each client signs the masked gradient by leveraging the modified Short Group Signature (SGS) [28] scheme to generate a signature message instantiated with cryptographic primitives **SIG.sign** as Equation (4). The details are provided in **Algorithm 1**.

$$\sigma_{G_i^{t+1}} \leftarrow \textbf{SIG.sign}\{sk_i^1, A_i, Msk\{G_i^{t+1}\}\}, \tag{4}$$

The client sets masked local gradients message as $LOC_i^{t+1} = (\sigma_{G_i^{t+1}}, Msk\{G_i^{t+1}\}, TC_i)$. Finally, each client forwards LOC_i^{t+1} to ED within its communication range. Note that the matrix $R_{s,i}$ has the same dimension as the local gradient matrix. Moreover, the gradients are typically 32-bit floating point values, which have to be scaled before masking. Prior work shows that a 16-bit wide fixed-point number is sufficient to achieve near lossless gradient quantization [34]. Therefore, we perform a mapping procedure that converts the gradients into integers within the range of 0 to 1024.

D. The Online-Aggregation Phase

After collecting enough messages, each ED verifies the masked gradients using cryptographic primitives **SIG.ver** $(\sigma, M) \rightarrow T/F$. Details are shown in **Algorithm 2**. If the authentication passes, the ED signs the gradients submitted by clients and forwards the message $Msg_{ed} = (\sigma_{ed}, Msk\left\{G_i^{t+1}\right\}, TC_{ed})$ to the RSU. The signature and verification protocol used in this phase is the same as the masking phase.

When the RSU receives messages from EDs, it first validates the legitimacy of the ED. Afterward, the RSU evaluates the quality of local gradients to identify malicious clients. Note that the identification process is optional. Finally, the RSU calculates the final global model as Equation (5).

$$Msk\{\psi^{t+1}\} = \sum_{i=1}^{\widetilde{U}} Msk\{G_i^{t+1}\}, \tag{5}$$

where $\widetilde{U}$ is the set of clients that passed the verification conducted by RSUs. The process will be repeated until the FL training output converges. Afterward, RSU performs the unmasking process and calculates the latest global model ψ^{t+1} as shown below:

$$\psi^{t+1} = (Msk\{\psi^{t+1}\} - \sum_{i=1}^{|\widetilde{U}|} R_{r,i})/|\widetilde{U}|. \tag{6}$$

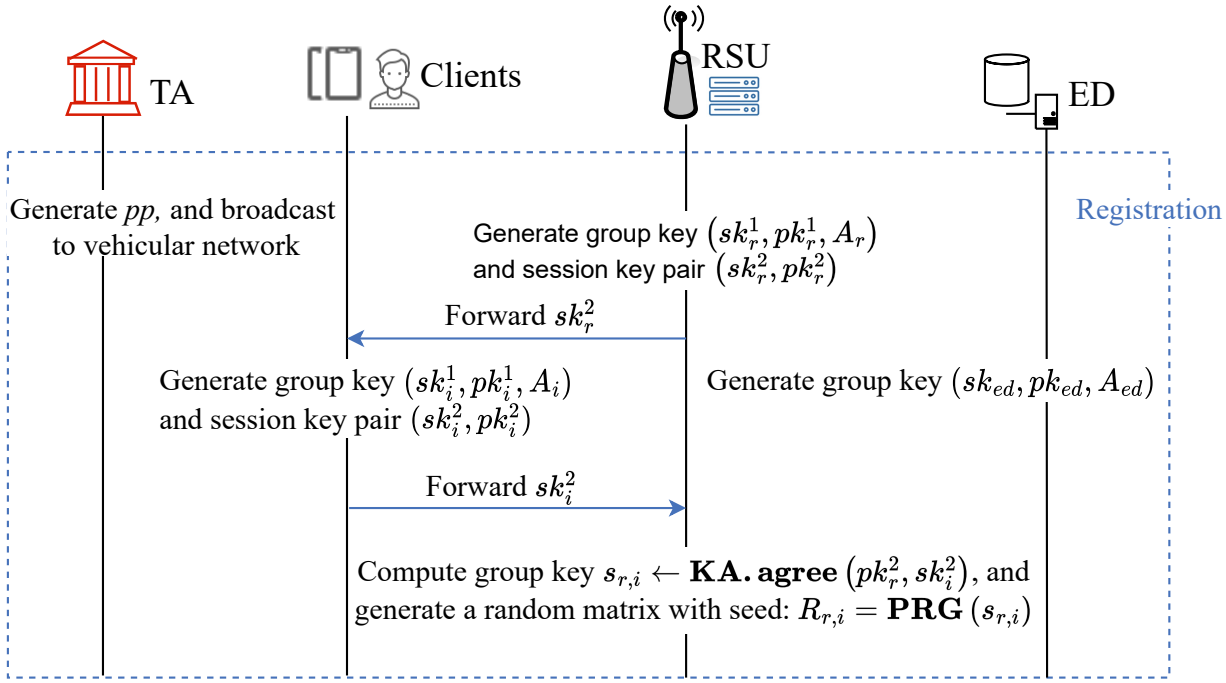


Fig. 2. Privacy-preserving FL aggregation protocol–Registration phase.

The random matrix $R_{r,i} = \mathbf{PRG}(s_{r,i})$ is generated in the registration phase. It is obvious that the session keys calculated by client i and RSU respectively are equal and satisfy the following equations.

Client i calculates:

$$s_{r,i} = sk_r^2 \cdot pk_i^2 = x_r \cdot x_i g. \tag{7}$$

RSU calculates:

$$s_{i,r} = sk_i^2 \cdot pk_r^2 = x_i \cdot x_r g = x_r \cdot x_i g. \tag{8}$$

Finally, RSU forwards the latest global model ψ^{t+1} to each client in $\widetilde{U}$. The detailed specification of this phase is provided in Fig.3. We stress that EDs cannot get any private information but the masked gradients.

E. The Continuous Authentication.

The continuous authentication method is developed to choose reliable clients and block malicious clients. The method includes three steps that could be executed in parallel with the aggregation procedure.

ED authentication. Each vehicle signs the gradients and transmits them to the ED. Upon receiving the message, the ED verifies the legitimacy of the vehicle, the integration, and the correctness of the message.

RSU authentication. Once the authentication conducted by the ED is successful, the ED will sign the validated gradients and send them to the RSU. The RSU will verify the legitimacy of the EDs, the integration, and the correctness of the message. Furthermore, the RSU evaluates the quality of local gradients to identify malicious clients.

Malicious client revocation. If the RSU detects any malicious client, it shall promptly report to the TA. The TA will block the malicious client by revoking its public key. Afterward, the malicious client cannot pass the authentication conducted either by EDs or RSUs.

TABLE I
SYMBOL TABLE AND DESCRIPTIONS

Stage	Symbol	Description
Registration	$(E, \mathbb{G}, g, q, h, \mu, v, \Gamma)$	System public parameters
	(sk_i^1, pk_i^1, A_i)	Group secret key/public key of client i
	(sk_i^2, pk_i^2)	Session secret key/public key of client i
	(sk_r^1, pk_r^1, A_r)	Group secret key/public key of RSU
	(sk_r^2, pk_r^2)	Session secret key/public key of RSU
	sk_{ed}, pk_{ed}	Secret key/public key of ED
	$H(X)$	Hash function $H : \mathbb{Z}_p^* \leftarrow \{0,1\}^*$
Online-Masking	TC	Timestamp
	σ	Signature generated by **SIG.sign**
	$s_{i,j}$	Session key generated by **KA.agree**
	$R_{i,r}$	Random number generated by **PRG**
	ψ^t	Global model at iteration t
	G_i^t	Local gradient matrix of client i at iteration t
Online-Aggregation	$Msk\{G_i^t\}$	Masked local gradient matrix
	Msg_{ed}	Signed message of ED
	ψ_{Msk}^{t+1}	Masked global gradient matrix
	U	Set of clients that have passed ED authentication
	$\widetilde{U}$	Set of clients that have passed RSU authentication
	ψ^{t+1}	The global model at iteration $t+1$

VI. Security and Privacy Analysis

In this section, we formally analyze the security and privacy properties of our scheme.

A. Security Analysis

Here, we demonstrate that our scheme is a secure aggregation protocol against active adversaries and semi-honest servers. The active adversaries [35] mean parties that deviate from this scheme, sending incorrect and/or arbitrarily chosen messages to other clients and sharing the keys with others, while the semi-honest servers are EDs defined in Section IV.C.

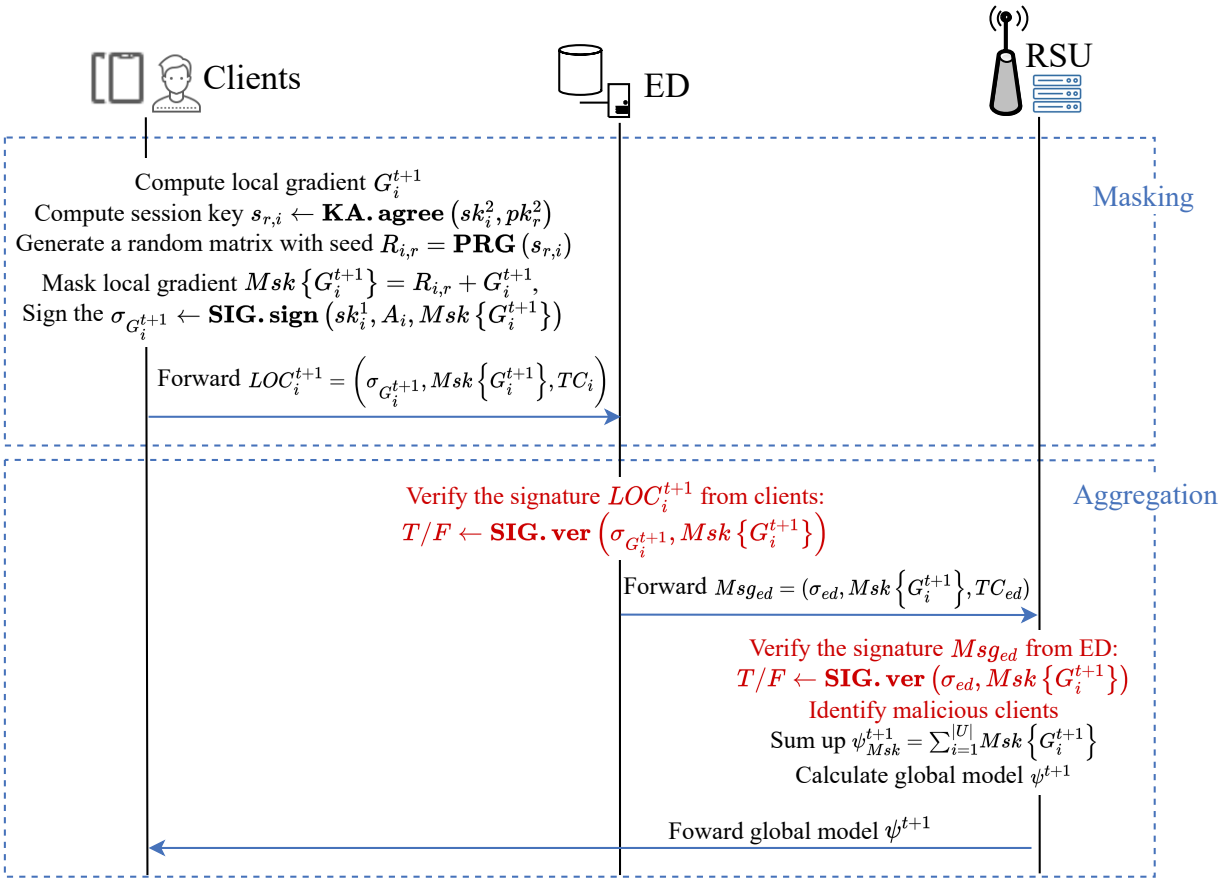


Fig. 3. Privacy-preserving FL aggregation protocol–Masking and aggregation phase.

Algorithm 1 SIG.sign(sk_i, A_i, M)

Require: $pp = (E, \mathbb{G}_1, g, h, \mu, \nu, \Gamma)$, client's key pairs (sk_i, pk_i, A_i) and encrypted local gradient M.

for all clients i with identity ID_i **do**
  Select random numbers $a, b \in \mathbb{Z}_p^*$
  Set $T_1 = a\mu, T_2 = b\nu, T_3 = A_i + (a+b)h$
  Set $\alpha = ax_i, \beta = bx_i$
  Select random numbers $c_a, c_b, c_x, c_\alpha, c_\beta \in \mathbb{Z}_p^*$
  Set $C_1 = c_a\mu, C_2 = c_b\nu$, $C_3 = e(T_3, g)^{c_x} e(h, (-c_a - c_b)\gamma + (-c_\alpha - c_\beta)g)$, $C_4 = c_x T_1 - c_\alpha \mu$, $C_5 = c_x T_2 - c_\beta \nu$
  Set $\varphi = H(M||T_1||T_2||T_3||C_1||C_2||C_3||C_4 ||C_5||TC_i)$
  Set $s_a = c_a + a\varphi, s_b = c_b + b\varphi, s_x = c_x + x_i\varphi$, $s_\alpha = c_\alpha + \alpha\varphi, s_\beta = c_\beta + \beta\varphi$
  $\sigma_i = (T_1, T_2, T_3, C_3, \varphi, s_a, s_b, s_x, s_\alpha, s_\beta)$
  return $Msg = (\sigma_i, pk_i, M, TC_i)$
end for

1) For Attack Model $\mathcal{A}_1$: For the attack model, $\mathcal{A}_1$ occurred between clients and EDs, we set the ED $\mathcal{S}$ as "honest-but-curious". Moreover, all received gradient data is encrypted, even if $\mathcal{S}$ is curious about one certain real gradient $\mathcal{X}$ from encrypted data $\widetilde{\mathcal{X}}$. To clarify the security of our scheme in gradient transmission between clients and EDs, we define a

Algorithm 2 SIG.ver(σ, M)

Require: Collect $Msg_1, Msg_2, ..., Msg_k$ from all clients.

for $i = 1$ to k **do**
  Set $\widetilde{C_{1,i}} = -\varphi_i T_{1,i} + s_{a,i}\mu$ ▷$C_{1,i}$ represents the variable C_1 for client i.
  Set $\widetilde{C_{2,i}} = -\varphi_i T_{2,i} + s_{b,i}\nu$
  Set $\widetilde{C_{4,i}} = s_{x,i} T_{1,i} - s_{\alpha,i}\mu$
  Set $\widetilde{C_{5,i}} = s_{x,i} T_{2,i} - s_{\beta,i}\nu$
  if $\varphi_i \neq H(M||T_{1,i}||T_{2,i}||T_{3,i}||\widetilde{C_{1,i}}||\widetilde{C_{2,i}}||C_{3,i}||\widetilde{C_{4,i}} ||C_{5,i}||TC_i)$ **then**
    Reject client i.
  end if
end for
Set $\prod_{i=1}^n \widetilde{C_{3,i}}^{\delta_i} = e(\sum_{i=1}^n \delta_i(s_{x,i}T_{3,i} - (s_{\alpha,i} + s_{\beta,i})h - \varphi_i g), g) e(\sum_{i=1}^n \delta_i(\varphi_i T_{3,i} - (s_{a,i} + s_{b,i})h), \Gamma)$
if $\prod_{i=1}^n C_{3,i}^{\delta_i} \stackrel{?}{=} \prod_{i=1}^n \widetilde{C_{3,i}}^{\delta_i}$ **then**
  Accept $Msg_1, Msg_2, ..., Msg_k$.
else
  Reject $Msg_1, Msg_2, ..., Msg_k$.
end if

Gradient Secure Computing Theory.

Definition 1 (Gradient Secure Computing Theory). *For the encrypted gradient set $\mathcal{Q} = \{\widetilde{\mathcal{X}}_1, \widetilde{\mathcal{X}}_2, \cdots, \widetilde{\mathcal{X}}_n\}$, Gradient Secure Computing Theory requires that any data $\widetilde{\mathcal{X}}$ provided to $\mathcal{S}$ satisfies the following properties for $\mathcal{X}$: existing a completely random number $\mathcal{R}$ makes equation $\widetilde{\mathcal{X}} = mask(\mathcal{X}, \mathcal{R})$ hold.*

The theory presents the encryption requirements of gradient $\mathcal{X}$. That is to say, any received data $\widetilde{\mathcal{X}}$ by $\mathcal{S}$ cannot map to any $\mathcal{X}$. Therefore, $\mathcal{S}$ cannot learn more knowledge with respect to $\mathcal{X}$ from $\widetilde{\mathcal{X}}$. In our proposal, through **Algorithm 2**, the ED only obtains masked gradients $\{Msk\{G_i^t\}|1 \leq i \leq |U|\}$, where the mask is $\mathcal{R}$ generated by a secure Pseudorandom Generator (PRG).

It is worth mentioning that the security of $\widetilde{\mathcal{X}}$ under attack model $\mathcal{A}_1$ depends on the reliability of the constructed random number $\mathcal{R}$, that is, EDs cannot break ECDHP assumption to compute session key s. Therefore, our proposal satisfies the theory, which effectively guarantees that any ED can not obtain real gradient information from received messages. For attack model $\mathcal{A}_1$ occurred between clients and EDs, our proposal is secure.

2) For Attack Model $\mathcal{A}_2$: For attack model $\mathcal{A}_2$, collusion may arise between clients and EDs. We conduct specific security analyses for the above scenario. We assume ED $\mathcal{S}$ and clients will cooperate to steal others' data. ED $\mathcal{S}$ receives n masked gradient $\{\widetilde{\mathcal{X}}_1, \widetilde{\mathcal{X}}_2, \cdots, \widetilde{\mathcal{X}}_n\}$. Supposing client ξ $(0 \leq \xi \leq n-1)$ conspires with the $\mathcal{S}$, all the information about gradient data that $\mathcal{S}$ can collect is $\{\mathcal{X}_1, \mathcal{X}_2, \cdots, \mathcal{X}_\xi, \widetilde{\mathcal{X}}_{\xi+1}, \cdots, \widetilde{\mathcal{X}}_n\}$. However, the encrypted data $\widetilde{\mathcal{X}}$ is independently generated by computing $\widetilde{\mathcal{X}} = mask(\mathcal{X}, \mathcal{R}) = \mathcal{X} + \mathcal{R}$ with random number $\mathcal{R}$. The ED $\mathcal{S}$ and colluded clients cannot obtain $\mathcal{R}$ without breaking the ECDHP assumption to compute session key s. Therefore, our proposal still satisfies Gradient Secure Computing Theory in this case, which has effectively guaranteed that any ED and colluded clients cannot learn more knowledge about real gradient information from $\{\widetilde{\mathcal{X}}_{\xi+1}, \cdots, \widetilde{\mathcal{X}}_n\}$. For attack model $\mathcal{A}_2$ occurred between clients and EDs, our proposal is secure.

B. Privacy Analysis

In this section, we discuss our privacy properties against semi-malicious servers and active adversaries. It is worthy of note that our scheme can preserve multiple private attributes such as anonymity and unlinkability.

Anonymity: The property of anonymity is enhanced since no identity information is revealed during the aggregation process. Let us give a proof sketch for anonymity. As a matter of fact, throughout the entire process of FL training, ID_i has never been transmitted as an integral component of the message by any node at any phase. Moreover, the authentication protocol established under a modified SGS [28] satisfies the zero-knowledge properties, making it impossible to infer ID_i based on the message it transmits. Solely TA possess knowledge of the ID_i. Suppose an active adversary can break anonymity, then it can have the identity ID_i of node i, which contradicts the assumption that TA is a trusted authority.

Unlinkabiltiy: Due to the distinct parameters present in the messages, denoted as $Loc = (\sigma_i^2, y_{i,t}, t)$ and $Msg = (\sigma_i^1, pk_i, TC)$ even if they originate from the same sender, the likelihood of being tracked by an attacker is diminished. Thus, our scheme provides unlinkability so that no attacker can tell whether two messages were derived from the same sender.

VII. Experiments

In this section, we first analyze the complexity of the proposed scheme and related schemes. Secondly, we provide the necessary experimental Settings, evaluation methods of the models, and the latest methods that need to be analyzed and compared. We then verify the validity of our approach by evaluating its performance on two common datasets.

A. Complexity analysis

We analyze the performance of the protocol, comparing with the computational and communication costs among methods such as TJL [36], Secure Aggregation [11], and BatchCrypt [37]. The complexity analysis of all schemes in Table II is evaluated by assuming a single aggregator and n clients per training round, where each client holds a data vector of size m. Compared to these schemes, our approach achieves higher efficiency in terms of computational and communication costs. It is worth noting that our approach requires only one round of communication, which is superior to all existing aggregation schemes.

- Client (Vehicle)
  - **Computation cost**: $\mathcal{O}(n)$. Each vehicle's computation cost can be broken up as 1) local dataset training, 2) masking the gradients G, and 3) signing the gradients G.
  - **Communication cost**: $\mathcal{O}(m)$. The communication cost of each vehicle is sending a signed data vector of size m+181B to the aggregator (181B is the signature size).
- Aggregator (ED+RSU)
  - **Computation cost**: $\mathcal{O}(mn)$. The computation cost can be broken up as 1) ED Verifies the signatures, which is $\mathcal{O}(mn)$, 2) RSU sums up the masked gradient and calculates the latest global model, which takes time $\mathcal{O}(1)$.
  - **Communication cost**: $\mathcal{O}(n)$. The RSU's communication cost is sending the new global model to the vehicles, which is $\mathcal{O}(n)$ in total.
    We have excluded the communication cost between the ED and RSU from our calculations, as the wired communication method used between them has a negligible impact.

B. End-to-End Benchmarks

We evaluate end-to-end system performance for two tasks on two datasets of MNIST and CIFAR100. Table 2 provides an overview of the tasks and models we use in our evaluation. All the experiments were conducted on a GPU server with

TABLE II
COMPUTATION, COMMUNICATION, AND COMMUNICATION ROUNDS (BETWEEN SERVER AND CLIENTS) PER TRAINING ROUND

Approach	Computation (Server)	Communication (Server)	Computation (Client)	Communication (Client)
Secure Aggregation [11]	$\mathcal{O}(mn^2)$	$\mathcal{O}(mn+n^2)$	$\mathcal{O}(mn+n^2)$	$\mathcal{O}(m+n)$
TJL [36]	$\mathcal{O}(mn)$	$\mathcal{O}(mn)$	$\mathcal{O}(m^2)$	$\mathcal{O}(mn)$
BatchCrypt [37]	$\mathcal{O}(m\log^2 n)$	$\mathcal{O}(mn)$	$\mathcal{O}(m\log^2 n)$	$\mathcal{O}(m\log n)$
This Work	$\mathcal{O}(mn)$	$\mathcal{O}(n)$	$\mathcal{O}(m)$	$\mathcal{O}(m)$

n: the total number of local clients
m: the length of local gradients

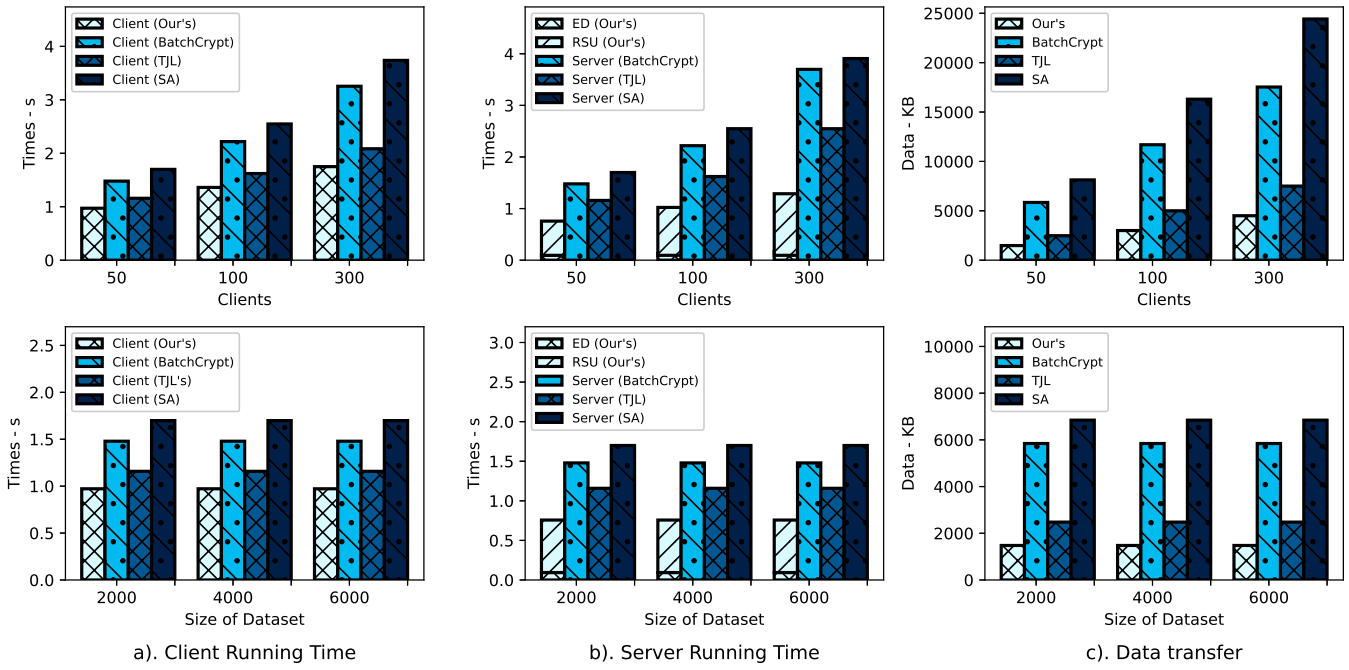


Fig. 4. The wall-clock running time (a,b) and data transfer (c). The measurement is conducted using our protocol and [37] [11] [36](only the online phase time is shown). When varying the number of clients, we fix the input dataset size m = 2000 and when varying the dataset size we fix the number of clients n = 50. Bars represent the average value for 10 rounds, which represents the time required by either the client or server for one round of training.

Nvidia 3090 graphics card, 128GB Memory, and Xeon 5218R CPU.

1) Setting of Dataset :

- The MNIST dataset is a widely used benchmark dataset in the field of machine learning and computer vision. It consists of a collection of 60,000 grayscale images of handwritten digits (0-9) for training, along with an additional 10,000 images for testing. Each image is a 28x28 pixel square, representing a digit drawn by various individuals.
- The CIFAR-100 dataset is a popular benchmark dataset in the field of computer vision and image recognition. It consists of 60,000 color images divided into 100 classes, with each class representing a specific object or concept. The dataset is split into 50,000 training images and 10,000 test images. Each image is a 32x32 pixel square, containing objects from various categories such as animals, vehicles, household items, and natural scenes.

2) Setting of Models: Considering the differing scales of the two datasets, we used different networks tailored to each dataset. For the relatively simple images in the MNIST, we utilized a CNN network with three convolutional layers. The convolutional sizes increased progressively from 16 to 32 and then 64. Following each convolutional layer, a 2x2 pooling layer was applied. For the more complex CIFAR-100, we employed ResNet-34 as the backbone network. ResNet-34 consists of 34 convolutional layers, and after the convolutional layers, the output is passed through a fully connected layer with 100 units for final classification. A summary of the model's parameters is provided in Table III.

3) Comparison of Our Protocol: We now show the performance compared with three state-of-the-art schemes which are BatchCrypt [37], SA [11], and TJL [36]. They are all privacy-preserving aggregation Schemes designed for FL suits for IoT tasks. First, we conducted multi-experiments to assess the running time on both clients and servers. Following this, we evaluated the data transfer from clients to servers with different data scales and numbers of clients. To better evaluate the performance, the number of clients is varying within $n = \{50, 100, 300\}$. We also use different batch sizes for

TABLE III
SUMMARY OF MODELS

	MNIST	CIFAR100
Backbone	CNN	ResNet34
Input Size	1x28x28	3x224x224
Output Units	10	100
Parameters	70,282	21,335,972
Gradients Size	280KB	82MB

clients' inputs $m = \{2000, 4000, 6000\}$.

Running time for clients. We plot the wall-clock running time for different scales of clients in Fig. 4a. Our protocol performs better for running time with the increasing number of clients and dataset. Additionally, our protocol scales better concerning the increasing number of clients (i.e., our solution is 1×1.4 faster than TJL with 50 clients, 1×1.6 faster with 100 clients, and 1×2 faster with 300 clients.).

Running time for the servers (EDs and RSUs). We plot the wall-clock running time for the server. In general, our cost is lower than the other three schemes shown in Fig.4b. It can be seen that the time required for a round of training grows linear with the number of clients, while positively stable with respect to the size of dataset. Additionally, in our approach, EDs are only responsible for data integrity verification, while most of the time-consuming operations are handled by the trusted RSU (i.e., our solution is 1×1.4 faster than SA with 50 clients, 1×1.8 faster with 100 clients, and 1×2.5 faster with 300 clients.).

Data transfer. We evaluate the data transfer, which is the size of data sent by the client to the server. As seen in Fig.4c, the data transferred from the clients to the RSUs grows linearly with the number of clients increasing. While the data transmitted by the client to RSU is independent of the size of dataset. We can conclude that the size of client's dataset has little impact on communication costs. This is mainly because instead of encrypting the local gradients (BatchCrypt, TJL), our scheme utilizes masking to protect the gradients (i.e., our solution is 1×1.7 faster than BatchCrypt with 50 clients, 1×1.7 faster with 100 clients, and 1×7 faster with 300 clients.).

Summary. Our protocol outperforms the other three schemes. Our approach results in varying degrees of reduction in the running time for both the client and server, with the most significant improvement observed in larger networks. Additionally, our protocol demonstrates better data transfer performance in the presence of a large number of clients and extensive input dimensions.

Effectiveness of our protocol. As shown in Fig.5, we have measured the total time required for one complete learning round including Registration Phase and Online Phase. The results indicate a total training round needs about 1.7s for MNIST and 16s for CIFAR100. During the Online Phase, EDs verify signatures from clients and sign the validated gradients, while RSU performs operations such as signature verification, identifying malicious clients, and gradient aggregation. The clients, on the other hand, carry out

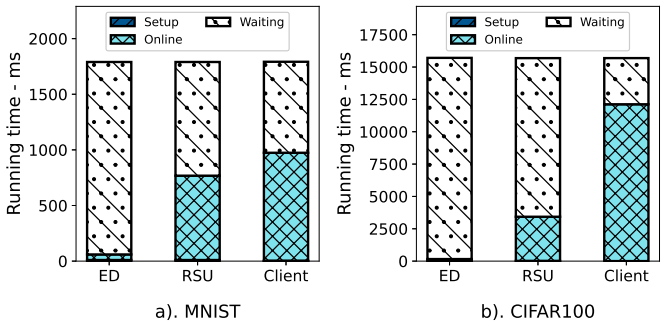


Fig. 5. Breakdown of running time.

operations such as gradient masking and signature generation. It's important to note that the time required for model masking and signature generation varies depending on the size of the model. To further evaluate the practicality of our protocol, we measure the time required for one client and server varying with $n = 100, 500, 1000$rounds in Table IV. The running time of Registration Phase becomes negligible after conducting a sufficient number of FL rounds. That's because, for each FL training task, the Registration Phase is executed only once, while the online phase is performed for every round of FL.

Training to coverage. Rather than analyzing a single training round, we assess end-to-end performance by training two benchmark datasets until they reach convergence. Fig. 6 shows the different training processes of each dataset, including the accuracy and loss curves. The training process of CIFAR100 is shown in Fig. 6a, the model did not converge until the 15th round of training. The reason is that the Resnet model is too complex and the input image is stretched into 224x224x3, leading to the feature explosion. In contrast, in Fig. 6b, MNIST is much simpler and the CNN model has reached the convergence state in the 3rd round of training. Moreover, due to the exactitude of the experiment, we conducted multiple rounds of training on both datasets and documented the highest and lowest levels of accuracy attained.

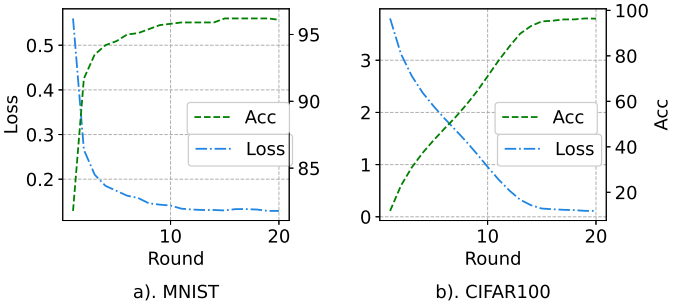


Fig. 6. Accuracy of FL with CIFAR and MNSIT.

C. Real Applications in VANETs

We have evaluated and compared our proposed scheme using two common machine-learning datasets. To further in-

TABLE IV
RUNNING TIME PER CLIENT WITH DIFFERENT FL ROUNDS

FL Rounds	Server	Client	Total Time	Registration Time
100	6116.88 ms	7843.81.57 ms	13960.69 ms	154.18 ms (1.1%)
500	29585.77 ms	37938.47 ms	67524.12 ms	154.18 ms (0.2%)
1000	59254.79 ms	75983.72 ms	135238.52 ms	154.18 ms (0.1%)

vestigate the performance of our proposed scheme in the real world, we implement two vehicular applications using our proposed protocols.

1) Setting of Applications:

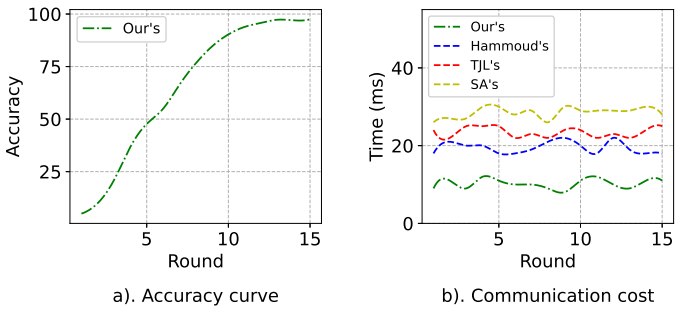
- Traffic-sign Recognition (TSR): TSR is a technology that uses computer vision algorithms to detect and interpret traffic signs in real-time. The TSR system captures an image of the traffic sign and identifies its shape, color, and content. The system can then provide feedback to the driver or autonomous vehicle, such as a visual alert or audible warning, to remind them to follow the instructions on the sign. TSR technology can improve road safety by reducing the risk of accidents caused by drivers or autonomous vehicles failing to recognize or follow traffic signs, and it can also improve traffic flow by providing more accurate and timely information to drivers.
- Traffic-flow Prediction (TFP): TFP is the process of forecasting the volume, speed, and other characteristics of vehicular traffic at a future time. It uses historical and real-time traffic data, as well as other relevant information such as weather and events. The ultimate goal of traffic flow prediction is to minimize congestion and optimize traffic flow.

2) Traffic-sign Recognition: In this task, we chose German Traffic Sign Recognition (GTSRB) [38], which comprises over 50,000 images of 43 frequently encountered traffic signs. We utilized a fundamental CNN architecture with three 5x5 convolutional layers, utilizing 39,000 images for training and 12,000 images for testing. In each training round, the clients update the local model with a learning rate of 0.01, batch size of 64, and epochs of 32.

Training accuracy. As seen in Fig.7a, our protocol achieves an accuracy close to 80% after 10 iterations of training, and it is observed that the model has not yet converged. The experimental results demonstrate that after 15 training rounds, our model has converged with an accuracy of over 95%.

Communication cost. In order to demonstrate the application effectiveness of our solution in VANETs, we have introduced a new comparative approach, which is Hammoud's [16]. The comparison of the communication cost between clients and servers is shown in Fig.7b. We conclude that our scheme outperforms the other protocols, and requires significantly less communication cost.

3) Traffic-flow Prediction: To make predictions in this task, we utilize real traffic patterns derived from the PeMS (Performance Measurement System) dataset [39]. This dataset is collected by the California Department of Transportation in the United States and includes traffic data from loop detectors installed on highways in the Los Angeles metropolitan area. It



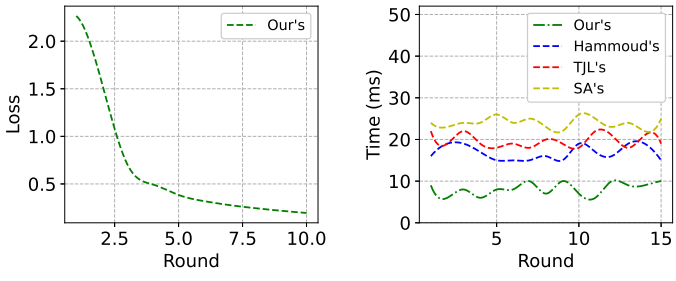
a). Accuracy curve b). Communication cost

Fig. 7. Comparison on Traffic-sign Recognition.

provides hourly aggregated traffic data for over 1,000 sensor locations, such as traffic flow, speed, and occupancy. We then develop a GCN (Graph Neural Network) model for prediction and train it using a learning rate of 0.01, a batch size of 64, and 32 epochs in each training iteration.

Training accuracy. The performance of our approach in regression tasks is demonstrated in Fig.8. The loss curve of the model after 20 training iterations is illustrated in Fig.8a. These results indicate that our scheme performs well in traffic-flow prediction. It is worth mentioning that we also compared the two real datasets mentioned above with recent studies, and our experimental results are described below.

Communication cost. To further evaluate the performance of our protocol, we plot the communication cost results in Fig.8b. It shows that our method performs comparably to other schemes. The communication cost fluctuates around 10 ms, which meets the requirement for VANETs. The reliability and practicality of our protocol are confirmed through the communication cost.



a). Loss curve b). Communication cost

Fig. 8. Comparison on Traffic-flow Prediction.

VIII. Conclusions

In this paper, we systematically study utilizing secure aggregation to FL in VANETs. We survey existing techniques in FL and highlight the stringent efficiency and privacy requirements. Therefore, we propose a novel aggregation protocol that provides security in two attack models. Our constructions show how to enhance privacy with a continuous authentication implementation. Moreover, our scheme makes use of the edge device's computational capacity to authenticate the FL clients, which reduces the burden of the RSUs. The simulations show that the running time of clients and servers outperforms the existing schemes. Meanwhile, with real-world applications, we evaluated the communication cost which fluctuates around 10 ms, meeting the requirement of VANETs. Therefore, we affirm the reliability and practicality of our scheme. Future work would focus on reducing the overall transfer of data by compressing the encrypted gradients.

References

[1] H. Xiao, J. Zhao, Q. Pei, J. Feng, L. Liu, and W. Shi, "Vehicle selection and resource optimization for federated learning in vehicular edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11 073–11 087, 2021.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[3] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in neural information processing systems*, vol. 30, 2017.

[4] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 739–753.

[5] J. Zhang, B. Chen, X. Cheng, H. T. T. Binh, and S. Yu, "Poisongan: Generative poisoning attacks against federated learning in edge computing systems," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3310–3322, 2020.

[6] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," in *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*. Internet Soc, 2018.

[7] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning," in *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, 2017, pp. 603–618.

[8] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*. IEEE, 2017, pp. 3–18.

[9] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[10] Y. Zhao, J. Zhao, M. Yang, T. Wang, N. Wang, L. Lyu, D. Niyato, and K.-Y. Lam, "Local differential privacy-based federated learning for internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8836–8853, 2020.

[11] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[12] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "Verifynet: Secure and verifiable federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 911–926, 2019.

[13] C. Hahn, H. Kim, M. Kim, and J. Hur, "Versa: Verifiable secure aggregation for cross-device federated learning," *IEEE Transactions on Dependable and Secure Computing*, 2021.

[14] Y. Zhang and H. Yu, "Towards verifiable federated learning," *arXiv preprint arXiv:2202.08310*, 2022.

[15] A. Wainakh, A. S. Guinea, T. Grube, and M. Mühlhäuser, "Enhancing privacy via hierarchical federated learning," in *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 2020, pp. 344–347.

[16] A. Hammoud, H. Otrok, A. Mourad, and Z. Dziong, "On demand fog federations for horizontal federated learning in iov," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 3062–3075, 2022.

[17] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, B. McMahan *et al.*, "Towards federated learning at scale: System design," *Proceedings of machine learning and systems*, vol. 1, pp. 374–388, 2019.

[18] O. A. Wahab, A. Mourad, H. Otrok, and T. Taleb, "Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1342–1397, 2021.

[19] Q. Xia, W. Ye, Z. Tao, J. Wu, and Q. Li, "A survey of federated learning for edge computing: Research problems and solutions," *High-Confidence Computing*, vol. 1, no. 1, p. 100008, 2021.

[20] S. R. Pokhrel and J. Choi, "Federated learning with blockchain for autonomous vehicles: Analysis and design challenges," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4734–4746, 2020.

[21] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Differentially private asynchronous federated learning for mobile edge computing in urban informatics," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2134–2143, 2019.

[22] F. Yu, H. Lin, X. Wang, A. Yassine, and M. S. Hossain, "Blockchain-empowered secure federated learning system: Architecture and applications," *Computer Communications*, vol. 196, pp. 55–65, 2022.

[23] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 700–10 714, 2019.

[24] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Transactions on Communications*, vol. 68, no. 2, pp. 1146–1159, 2019.

[25] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time optimization for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2457–2471, 2020.

[26] X. Hu, G. Wang, L. Jiang, S. Ding, and X. He, "Towards efficient learning using double-layered federation based on traffic density for internet of vehicles," in *Web Information Systems and Applications: 18th International Conference, WISA 2021, Kaifeng, China, September 24–26, 2021, Proceedings*. Springer, 2021, pp. 287–298.

[27] M. A. Hossain, I. Elshafiey, and A. Al-Sanie, "Cooperative vehicle positioning with multi-sensor data fusion and vehicular communications," *Wireless Networks*, vol. 25, pp. 1403–1413, 2019.

[28] D. Boneh and X. Boyen, "Short signatures without random oracles," in *Advances in Cryptology-EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004. Proceedings 23*. Springer, 2004, pp. 56–73.

[29] P. Kushwaha, "Towards the equivalence of diffie-hellman problem and discrete logarithm problem for important elliptic curves used in practice," in *2017 ISEA Asia Security and Privacy (ISEASP)*. IEEE, 2017, pp. 1–4.

[30] A. Menezes, "Evaluation of security level of cryptography: the elliptic curve discrete logarithm problem (ecdlp)," *University of Waterloo*, vol. 14, 2001.

[31] C. Lyu, D. Gu, Y. Zeng, and P. Mohapatra, "Pba: Prediction-based authentication for vehicle-to-vehicle communications," *IEEE transactions on dependable and secure computing*, vol. 13, no. 1, pp. 71–83, 2015.

[32] M. Hellman, "New directions in cryptography," *IEEE transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[33] J. H. Bell, K. A. Bonawitz, A. Gascón, T. Lepoint, and M. Raykova, "Secure single-server aggregation with (poly) logarithmic overhead," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 1253–1269.

[34] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," in *International conference on machine learning*. PMLR, 2015, pp. 1737–1746.

[35] H. Fereidooni, S. Marchal, M. Miettinen, A. Mirhoseini, H. Möllering, T. D. Nguyen, P. Rieger, A.-R. Sadeghi, T. Schneider, H. Yalame *et al.*, "Safelearn: secure aggregation for private federated learning," in *2021 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2021, pp. 56–62.

[36] M. Mansouri, M. Önen, and W. Ben Jaballah, "Learning from failures: Secure and fault-tolerant aggregation for federated learning," in *Proceedings of the 38th Annual Computer Security Applications Conference*, 2022, pp. 146–158.
[37] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning," in *Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC 2020)*, 2020.
[38] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The german traffic sign recognition benchmark: A multi-class classification competition," in *The 2011 International Joint Conference on Neural Networks*, 2011, pp. 1453–1460.
[39] C. Chen, K. Petty, A. Skabardonis, P. Varaiya, and Z. Jia, "Freeway performance measurement system: mining loop detector data," *Transportation Research Record*, vol. 1748, no. 1, pp. 96–102, 2001.

Xia Feng received the B.S. degree in computer science and technology from Jiangsu University in 2008, and the Ph.D. degree in computer science and technology department from Anhui University in 2017. Her research interests include authentication protocols in IoT, blockchain, and applied cryptography.

Xiaofeng Wang is currently pursuing the M.E. degree in the School of Automotive and Traffic Engineering with Jiangsu University. His research interest includes authentication protocols, privacy security in IoV.

Haiyang Liu is working toward the M.E. degree in computer science and technology with the School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, China. His current research interests include blockchain security and cryptography in federated learning.

Haowei Yang received the B.E. degree in information security from Jiangsu University in 2021. His current research interests include vehicular ad-hoc network security and cryptography in federated learning.

Liangmeng Wang received the B.S. degree in computational mathematics from Jilin University, Changchun, China, in 1999, and the Ph.D. degree in cryptology from Xidian University, Xi an, China, in 2007. He is currently a Professor of Cyber Science and Engineering with Southeast University, Nanjing, 211189, China. He has published over 60 technical papers in international journals and conferences. His current research interests include security protocols and Internet of Things. He is a member of IEEE and ACM.