AEFL: Anonymous and Efficient Federated Learning in Vehicle Road Cooperation Systems with Augmented Intelligence of Things

Xiaoding Wang†, Jiadong Li†, Hui Lin*, Cheng Dai*, Sahil Garg, Georges Kaddoum

***Abstract*—As the Augmented Intelligence of Things (AIoT) advances within vehicle road coordination systems, challenges related to road traffic data transmission and processing are being increasingly addressed. However, this progress also brings significant risks of privacy data leakage. Federated Learning (FL), a distributed machine learning paradigm, effectively safeguards client data privacy by allowing multiple participants to collaboratively train models while keeping their data localized. Despite its benefits, FL faces challenges such as model parameter leakage and Byzantine attacks. To tackle these issues, this paper introduces an Anonymous and Efficient Federated Learning framework for vehicle-road coordination systems (AEFL), designed to ensure a secure and reliable vehicle data transmission process. This architecture incorporates a novel Group Pairing Onion Routing protocol, which leverages pairing cryptography principles for hierarchical data encryption. During the routing process, relay group nodes decrypt the corresponding layer, ensuring both data confidentiality and node anonymity. Additionally, a sampling method is proposed to accurately identify Byzantine vehicle nodes, enhancing the precision of FL aggregation without compromising overall model performance. Experimental results show that AEFL outperforms the classic TOR anonymous routing protocol, achieving a 100% message delivery rate more quickly. Under the same conditions, the anonymity of the source node and the destination node improves by 3.9% and 1.9%, respectively. When half of the nodes are compromised, path anonymity can be increased by 24.8%. Furthermore, our framework excels in federated learning aggregation efficiency, with a Byzantine adversary detection accuracy of up to 99%.**

***Index Terms*—Augmented Intelligence of Things, Federated Learning, Anonymous Routing, Byzantine Attacks, Privacy Protection.**

I. Introduction

IN recent years, the rapid development of vehicle road coordination systems has significantly improved data communication and processing efficiency between vehicles and road infrastructure [1]. These systems collect and transmit real-time traffic information about vehicles, locations, and areas, offering a promising solution for enhancing traffic safety and mobility. They play a crucial role in achieving intelligent urban traffic patterns by providing real-time road condition information, improving the driving experience, and strengthening vehicle safety measures [2]. To optimize the utilization of road traffic data and establish a secure and efficient transportation system, artificial intelligence (AI) technologies, such as deep learning, are increasingly utilized for training vehicle data models. Traditionally, machine learning tasks were performed on cloud servers or data centers, necessitating substantial network resources. However, with the rapid growth of vehicle data, offloading large amounts of data to remote servers may become impractical due to the required network resources and resulting latency. Additionally, from a privacy standpoint, vehicle clients are often reluctant to share their sensitive local data [3]. As a result, data training is transitioning from centralized data centers to numerous on-vehicle terminals. This shift leverages the Augmented Intelligence of Things (AIoT) in vehicle road systems to effectively address privacy and efficiency challenges in vehicular networks. Given the limitations of traditional in-vehicle network architecture in supporting the extensive coverage, large-scale connectivity, real-time processing, and intelligent computing required by the Internet of Things, AIoT [4], [5] has become a key trend in the evolution of future vehicle road collaboration systems. By integrating artificial intelligence technology with the Internet of Vehicles, AIoT provides smarter and more efficient solutions through the analysis and processing of vehicle network data. In recent years, AIoT has gained significant attention for its integration into these systems. Initially, various data types, including environmental, operational, business, and monitoring data, are collected in real-time through interconnected sensors. Subsequently, this data is intelligently processed and analyzed using data mining and machine learning techniques in vehicles, base stations, roadside units, or the cloud.

The concept of Federated Learning (FL) [6]–[9] has recently been introduced to establish AIoT systems in vehicle road cooperation. In FL, the training process is decentralized across client devices, eliminating the need to exchange data samples with a central server and sharing only model parameters to create a global model. This approach efficiently utilizes

†These authors contributed equally.
*Corresponding authors.

X. Wang, J. Li and H. Lin are with the Fujian Provincial Key Laboratory of Network Security and Cryptology, College of Computer and Cyber Security, Fujian Normal University, Fuzhou, Fujian, 350117 P.R. China. (E-mail: wangdin1982@fjnu.edu.cn, ljddong1210@163.com, and linhui@fjnu.edu.cn)

Cheng Dai is with the college of Computer, Sichuan University, China (E-mail: daicheng@scu.edu.cn)

Sahil Garg is with the Electrical Engineering Department, École de technologie supérieure, Montréal, QC H3C 1K3, Canada and also with the Centre for Research Impact & Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, 140401, Punjab, India. (E-mail: sahil.garg@ieee.org)

Georges Kaddoum is with the Electrical Engineering Department, École de technologie supérieure, Montréal, QC H3C 1K3, Canada and also with the Artificial Intelligence & Cyber Systems Research Center, Lebanese American University, Beirut 03797751, Lebanon. (E-mail: georges.kaddoum@etsmtl.ca)

client computational resources and is recognized as a superior learning paradigm for enhancing privacy and communication efficiency [10]. When integrated into the AIoT framework for vehicle road cooperation systems, FL supports intelligent vehicle services such as autonomous driving, road safety prediction, and accurate vehicle detection. By facilitating collaborative learning between vehicles and roadside units, FL enhances privacy protection [11]. However, existing research highlights security challenges in FL related to model parameter leakage and Byzantine attacks.

Firstly, while Federated Learning (FL) clients train models locally without sharing data directly, malicious nodes can still infer user information by analyzing these local models [12]. To protect model parameters, strategies include preventing parameter leakage and concealing user identities. Common security methods such as Differential Privacy (DP) [13], Secure Multi-party Computation (MPC) [14], and Homomorphic Encryption (HE) [15] are used to enhance data privacy. However, research on effectively concealing user identities in FL is limited. Anonymous communication techniques like Onion Routing technology (Tor) [16], [17] utilize multi-layer encryption to establish anonymous channels, safeguarding user identities and offering potential privacy benefits in vehicle road cooperation systems.

Secondly, in federated learning, where each client can update the global model, vulnerabilities exist to Byzantine attacks from malicious users tampering with data or device malfunctions [18]. Defenses against such attacks typically involve comparing updates from multiple users and detecting anomalies [19]–[21]. However, a single error can significantly impact the model's performance [22]. These methods, which require extensive comparisons and computations, struggle to achieve a balance between robustness and efficiency, often falling short in practical, real-world scenarios. The challenge lies in effectively safeguarding user privacy while defending against Byzantine attacks in vehicle road cooperation systems.

To address the challenges mentioned above, this paper presents the Anonymous and Efficient Federated Learning framework for vehicle road cooperation systems (AEFL). This framework is specifically designed to mitigate significant security threats and ensure the secure and reliable operation of federated learning processes in vehicle road cooperation systems leveraging AIoT. The AEFL framework leverages onion routing technology and introduces a group pairing onion routing protocol (GPor) that allows nodes within the same group to efficiently encrypt or decrypt corresponding layers, facilitating data transmission. Model data encryption using pairing-based cryptographic methods ensures that intercepted data cannot be deciphered by attackers, ensuring node anonymity and data security.

Additionally, a sampling-based approach is introduced to effectively detect Byzantine adversaries in vehicles by computing the Euclidean distance across selected dimensions of each model vector. By excluding models from malicious vehicle nodes and computing the coordinate median of the remaining models, the accuracy of the aggregation process is enhanced. The key contributions of this paper are summarized as follows.

- We developed a group pairing onion routing protocol for vehicle road cooperation systems, which encrypts messages in layers along a predetermined path to protect the identity privacy of vehicle client nodes.
- We introduced a sampling-based Byzantine node detection algorithm to accurately identify malicious vehicle clients, improving model aggregation effectiveness and global model accuracy.
- We conducted a theoretical analysis and experimental evaluation of the AEFL framework, demonstrating its security and correctness. In terms of packet delivery rate, node anonymity, and path anonymity, AEFL achieves a 100% message delivery rate in a shorter time compared to the TOR protocol. Under the same conditions, the anonymity of the source and destination nodes is improved by up to 3.9% and 1.9%, respectively. Even when half of the nodes are compromised, the path anonymity can be improved by up to 24.8%. Additionally, the results show that our framework efficiently aggregates data and achieves a Byzantine adversary detection accuracy of up to 99% on the MNIST and CIFAR-10 datasets.

The rest of this paper is organized as follows: Section II reviews the related work. Section III outlines the system model and threat model. Section IV details the implementation of the model framework. Section V provides a theoretical analysis of the performance and security of the proposed framework. Section VI presents the experimental simulation results. Section VII concludes this paper and provides future research directions.

II. Related works

In this section, we will introduce the state-of-the-art technologies of federated learning in terms of communication anonymity and Byzantine attack resistance.

A. Anonymous Communication in Federated Learning

With escalating privacy concerns in federated learning, the development of robust privacy protection measures has become crucial for the architecture of FL systems. Lyu *et al.* [23] introduced the Fair and Privacy Preserving Deep Learning (FPPDL) framework, incorporating a three-layer onion encryption scheme to safeguard the privacy of individual model updates. Girgis *et al.* [24] devised a communication-efficient federated learning approach using an anonymous shuffling model to enhance privacy protection. Li *et al.* [25] designed an anonymity-centric privacy protection strategy for federated learning in autonomous driving, ensuring user anonymity through Zero-Knowledge Proofs (ZKP). Domingo-Ferrer *et al.* [26] developed a federated learning framework that ensures both privacy and security by achieving unlinkability between user identities and their updates through circulating updates among users before submission to the server. Additionally, Chen *et al.* [27] introduced the FedTor federated learning framework for the Internet of Things, providing user privacy guarantees through multi-layer encryption and reputation-based routing selection.

Anonymous routing has received significant attention in research for its ability to protect user privacy. Catalano *et al.* [28]

combined identity-based encryption with traditional public key encryption methods to enhance user privacy. Kotzanikolaou *et al.* [29] introduced BAR, a scalable anonymous communication system that integrates the broadcasting capabilities of dc-nets with layered encryption strategies from mix networks. Han *et al.* [30] developed a protocol for protecting the privacy of the source location through dynamic routing, effectively countering privacy leakage attacks while maintaining network longevity. Kim *et al.* [31] improved the security and privacy of Tor using Intel SGX in their SGXTor design, preventing code modifications and restricting data sharing with untrusted entities.

Despite these advancements, research on anonymous federated learning has largely overlooked challenges related to the privacy of vehicle client identities and routing security within vehicular cooperative networks.

B. Byzantine Attack Resistance in Federated Learning

In federated learning, Byzantine attacks occur when malicious entities intentionally participate in the federated learning process by transmitting incorrect gradients or models to disrupt it. These attackers can be part of the federated learning participants or the communication channels used for data or model transmission. Byzantine attacks can lead to deviations in the model's convergence, negatively impacting the accuracy and stability of the global model.

To undermine the global model, Byzantine adversaries typically replace model parameters with arbitrary values significantly different from those of a standard model. Mitigating Byzantine attacks focuses on designing a robust aggregation model that allows the server to exclude malicious gradients effectively. The Krum aggregation algorithm introduced in [22] selects the user gradient with the smallest sum of gradients among its "neighbors" as the global gradient. The Trimmed-Mean and Median algorithms presented in [18] decompose each user's gradient and use the average or median for aggregation, preventing the absorption of extreme opinions in the aggregation result. The Bulyan method proposed in [32] combines Krum and Trimmed-Mean approaches, utilizing Krum to select a subset of clients and then applying the Trimmed-Mean method for robust aggregation. The Multi-Krum method extended from Krum, as discussed in [22], selects several local models most similar to others and computes their average to generate the global model. In [18], random sampling for dimension reduction is initially employed, followed by spectral methods to remove malicious clients. The Sniper algorithm described in [33] constructs a graph by connecting vertices with small distances to form a connected subgraph and selects the largest subgraph for average aggregation. In [34], cosine similarity is utilized to aggregate user gradients with low similarity to the average gradient of all participants, excluding those deemed attackers, identifying malicious gradients based on gradient similarity.

These methods leverage statistical principles to estimate aggregated model updates. However, attackers may strategically design local models, selectively incorporating harmful parameters to bypass such defensive measures.

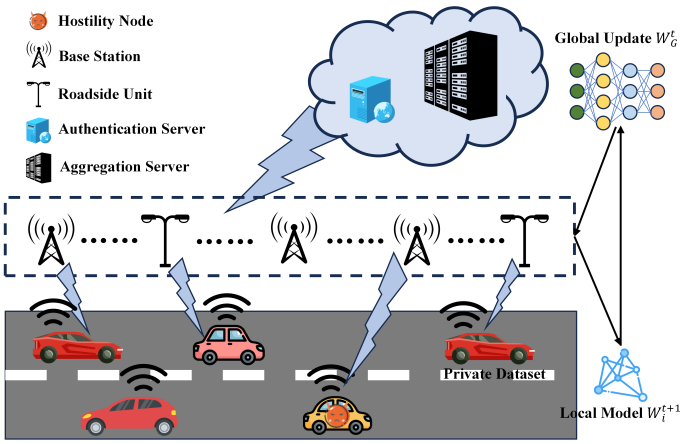

Fig. 1. Federated learning framework for vehicle road cooperation systems.

III. System Model and Threat Model

A. System Model

Fig. 1 illustrates the system model of the federated learning framework for vehicle road cooperation systems considered in this paper. In this system model, the vehicular cooperative network consists of multiple vehicle nodes, several base stations, roadside units, and two servers: the authentication server and the aggregation server. Each vehicle node within this network acts as a client in federated learning, responsible for training models on local data and sharing model updates. Base stations and roadside units function as onion routers, facilitating secure communication from vehicle nodes to the aggregation server by employing group pairing in onion routing to safeguard clients' model privacy. Prior to generating a shared key, the authentication server validates the onion routers, including base stations and roadside units. Upon network entry, an onion router must register with the authentication server, which stores registration details in a database and organizes the routers into groups. To verify an onion router, a client can request its registration information from the server. In each iteration, the aggregation server dispatches global parameters to selected clients, aggregates the received local parameters, and updates the global parameters.

B. Threat Model

To ensure the anonymity of communication during data transmission, our system addresses the following routing threats:

1) Node Identification Threat: We consider a scenario where an attacker controls multiple nodes across different groups within the anonymous network. The attacker's goal is to infer the locations of source and destination nodes by analyzing their previous and next hops.

2) Path Identification Threat: In this threat scenario, the attacker compromises specific nodes along the transmission path to analyze data flow through these nodes and deduce the entire transmission route.

3) Collusive Attack Threat: Collusive attacks involve compromised onion routers collaborating, both individually and

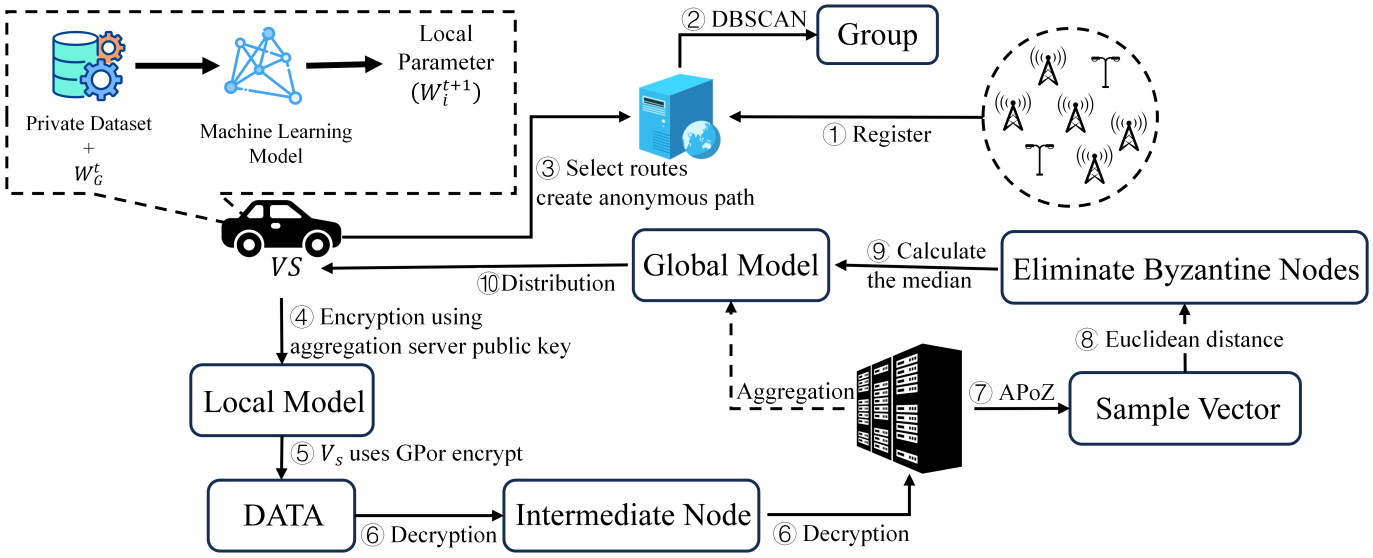

Fig. 2. Workflow of the AEFL framework.

collectively, to disrupt communication, posing significant privacy and security risks.

Additionally, we account for Byzantine adversaries aiming to introduce malicious parameters to disrupt the convergence of the global model. In each round, n clients are selected to upload their models, among which f are malicious. The proportion of malicious clients is limited by the condition $f < \frac{n}{2} - 1$.

In this paper, we consider both the communication process and the clients to be potentially untrustworthy. Therefore, our main objective is to protect user privacy and defend against Byzantine attacks.

IV. Implementation Details of the Proposed AEFL Framework

In this section, we will provide a detailed explanation of the proposed AEFL framework. The workflow of the AEFL framework is depicted in Fig. 2.

A. Overview of the Federated Learning Process in AEFL

Suppose there are N clients, each holding a private dataset $\mathcal{D}_i$ ($i \in [1, N]$). The federated learning process of the AEFL framework can generally be outlined as follows:

1) Global Model Distribution: If the global model is uninitialized, the aggregation server first sets the training tasks, defines the hyperparameters of the global model, and outlines the training process before initializing the global model as W_G^0. Subsequently, it selects n (where $n < N$) clients and distributes the global model W_G^t to them.

2) Local Model Training and Updating: Each client i replaces their local model with the global model W_G^t, denoted as W_i^t, where t is the current iteration index. The client then optimizes the local model using their private data as follows:

$$W_i^{t+1} = W_i^t - \eta \nabla \ell(W_i^t). \tag{1}$$

Here, η represents the local learning rate, and ℓ denotes the loss function.

3) Anonymous Data Packet Transmission: Subsequent to the previous procedure, the local models from different clients are transmitted to the aggregation server via an anonymous channel. Further details on this anonymous routing process are elaborated in Section IV-B.

4) Global Model Aggregation and Update: The aggregation server employs the FedAvg algorithm to merge the local models and update the global model to W_G^{t+1}, calculated as:

$$W_G^{t+1} = \frac{1}{n} \sum_{i=1}^{n} W_i^{t+1}. \tag{2}$$

Additionally, the Byzantine robust aggregation process of the global model utilized in the AEFL framework is further explained in Section IV-C.

B. Group Pairing Onion Routing Protocol

In vehicle road cooperation systems, ensuring the privacy of vehicle clients' local data and identities is crucial. Establishing secure anonymous communication is essential to prevent node tracking and conceal the locations of vehicles. To achieve this, secure anonymous channels must be set up before clients transmit data packets to the aggregation server. We employ the Group Pairing Onion Routing Protocol, equipped with robust authentication, authorization, and privacy-preserving encryption to thwart routing attacks. The Ad hoc On-Demand Distance Vector (AODV) serves as the basic network routing protocol for simplicity.

The steps for packet forwarding are as follows (see Fig. 2): Initially, k base stations and roadside units register with the authentication server to join the network (Step ①). The network is then topologically structured, and the DBSCAN unsupervised learning algorithm clusters nodes with similar characteristics into groups to promote mutual trust (Step ②). Subsequently, the vehicle client, acting as the source node

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3446725

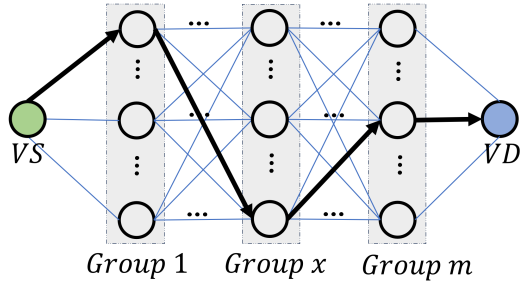

Fig. 3. GPor communication tunnel creation process.

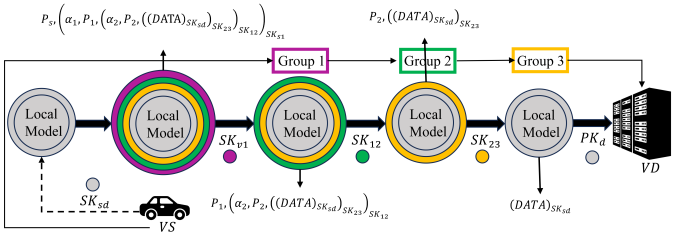

Fig. 4. GPor message encryption and decryption process.

VS, connects to the authentication server and selects an onion router group based on performance to establish an anonymous path (Step ③). The source node encrypts the local model using the public key of the destination node (Aggregation Server) to maintain confidentiality (Step ④). The GPor protocol then encrypts the message in layers as it passes through m groups, each containing an equal number of nodes (Step ⑤). The packets traverse this anonymous route, with each relay node decrypting its layer and forwarding the packet to the next node (Step ⑥). The transmission is deemed successful if the aggregation server decrypts the message within a specific timeframe; otherwise, it fails. Next, we elaborate on the encryption and decryption process within the GPor protocol.

Pairing onion routing is an anonymous communication protocol that combines onion routing with pairing-based cryptography principles. Similar to traditional onion routing, it employs multi-layer encryption and sequential decryption across network nodes to secure messages. Encryption keys for each layer are generated using elliptic curve cryptography and shared among adjacent nodes. This method obscures the message's origin and destination, making intercepted data challenging for attackers to decrypt, thereby preserving client privacy and security. In real-world road traffic networks, vehicle trajectories may vary, leading to unstable node connections that can degrade performance with direct pairing routing. To address this issue, we introduce "groups." As shown in Fig. 3, the group pairing onion routing protocol organizes network relay nodes, such as base stations and roadside units, into groups of size k/l, where l represents the number of nodes per group. Encrypted messages follow a predefined path through these groups, from VS to VD. Nodes within each group establish secure connections and share key privileges that enable any node to decrypt data received from the previous group and forward it to the next, enhancing data transmission efficiency.

Similar to pairing onion routing, in group pairing onion routing, the source node shares pairing keys with each group along the path. Fig. 4 illustrates the message encryption and decryption process in group pairing onion routing (using three relay nodes as an example), and we outline the shared key generation process for two groups $\{G_1, G_2\}$: i) If G_1 wishes to conceal its real identity from G_2, G_1 generates a pseudonym P_1, where $P_1 = \alpha_1 H_1(\mathrm{ID}_1)$, with α_1 being a random number from R_1. ii) G_1 generates a private key $PK_1 = MKH_1(\mathrm{ID}_1)$. iii) G_1 searches for G_2's identity ID_2 in the authentication server directory and computes $H_2(\mathrm{ID}_2)$. iv) G_1 calculates the shared key $SK_{12} = e(MKP_1, H_2) = e(H_1, H_2)^{MK\alpha_1}$ and sends the encrypted message along with the pseudonym P_1 to G_2. The process for G_2 to generate the shared key is similar. Upon receiving the message, the pseudonyms R_1, R_2, and the private key of R_2 are utilized to compute the shared key $SK_{12} = e(P_1, PK_2) = e(H_1, H_2)^{MK\alpha_1}$. The pseudocode for the encryption and decryption process of GPor is provided in Algorithm 1.

C. Byzantine Robust Secure Aggregation Protocol

Once vehicle clients securely transmit their local models to the aggregation server using the GPor protocol, the server combines these models into a global model. Existing Byzantine-robust aggregation methods often face high computational demands, leading to inefficiencies.

To address this challenge, we introduce a sampling-based aggregation scheme to reduce the computational load (see Fig. 2). Initially, we utilize the Average Percentage of Zeros (APoZ) algorithm [35] to assess the importance of each

Algorithm 1 Group Pairing Onion Routing Algorithm

Input: VS Local Parameter $DATA$.
Output: $DATA$ is securely sent to The Aggregation Server.

1: VS determines intermediate routing group $\{VS, G_1, G_2, \ldots, G_m, VD\}$.
2: VS selection random number $\{\alpha_s, \alpha_1, \alpha_2, \ldots, \alpha_m, \alpha_d\}$.
3: VS uses VD's private key to encrypts data as $DATA$.
4: VS generates a pseudonym P_s to hide the true identity by $P_s = a_s H_s(ID_s)$.
5: **Source node** VS **uses** $GPor$ **to encrypt** $DATA$**:**
6: **for** i from m to 1 **do**
7: VS computes $SK_{i,i+1} = e(MKP_i, H_{i+1}) = e(H_i, H_{i+1})^{MK\alpha_i}$ to encrypt $DATA = (\alpha_{i+1}, P_i, (DATA)SK_{i,i+1})$.
8: **end for**
9: VS sends $DATA$ to VD.
10: **Intermediate node decryption:**
11: **for** i from 1 to m **do**
12: G_i computes the shared key $SK_{i-1,i} = e(P_{i-1}, PK_i) = e(H_{i-1}, H_i)^{MK\alpha_{i-1}}$ to decrypt $DATA$ and forwards the decrypted message to the next hop.
13: **end for**

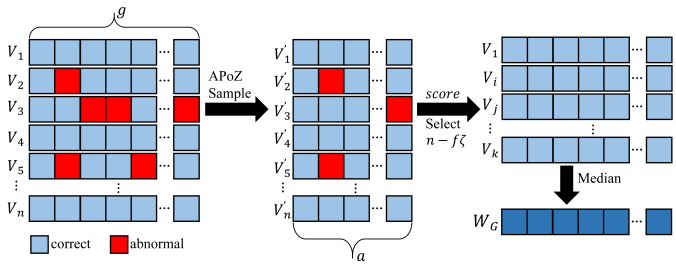

Fig. 5. Sampling process for identifying Byzantine nodes.

layer, focusing on dimensions that significantly influence the aggregation outcome (Step ⑦). This selection process aims to preserve the integrity of critical vector dimensions, thereby enhancing model accuracy. Subsequently, we measure the Euclidean distance between sampled vectors to identify potential adversaries, assuming that vectors distant from each other likely originate from Byzantine nodes (Step ⑧). Following the exclusion of these nodes, we compute the coordinate median of the remaining honest models (Step ⑨). This approach provides a more accurate representation of the optimal model by focusing on the median of vectors after eliminating malicious influences. Fig. 5 illustrates the sampling process for identifying Byzantine adversaries, where $V_1, V_2, \ldots, V_n$ represent the local model vectors of various clients, each vector containing g dimensions.

After receiving model updates from vehicle clients, the aggregation server follows the process outlined below. Firstly, we sample V_i using the APoZ algorithm to extract the a most important elements from the vector, resulting in the sampled vector V_i', where the elements are from the same dimensions. Secondly, we compute the Euclidean distance between each pair of V_i' and use a $score$ to represent the distance between the sampled vectors. Thirdly, we select the $n - f\zeta$ vectors with the smallest $score$ to form the subset S. These vectors, being close to each other, are considered correct, while the outlier vectors are excluded. Finally, we calculate the coordinate median (AGG) of the selected vectors to serve as the aggregation result. This entire process is detailed in Algorithm 2.

Algorithm 2 Byzantine Resistant Federated Learning Aggregation Algorithm

Input: Local Datasets $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n\}$.
Output: Global Model W_G^{t+1}.

Uploading models:
Client i trains local parameters $W_i^{t+1} = W_i^t - \eta \nabla \ell(W_i^t)$.
Sending local models to aggregation servers using $GPor$.
Aggregation Server uses the private key PK_d to decrypt the received model.
Aggregating models:
Use the $APoZ$ to select $m \in [0, a]$ elements from V_i recorded as V_i'.
for $i = 1$ to n **do**
    When $i \neq j$, $i \to j$ represents the distance from V_i' to V_j'.
    Compute the score$(i) = \sqrt{\sum_{i-j}(V_i' - V_j')^2}$.
end for
Select $Min = n - f\zeta$ vectors with the smallest $score$ to form the subset $S = \{V_i, \ldots, V_j\}$.
$W = AGG(V_i, \ldots, V_j)$.
Return W_G^{t+1}.

V. Performance and Security Analysis

In this section, we delve into a theoretical analysis of both the performance and security aspects of the AEFL framework. We scrutinize the anonymous routing algorithm, emphasizing key performance metrics such as the success rate of message forwarding and traceability. Additionally, we provide compelling evidence supporting the anonymity of source nodes, destination nodes, and data forwarding paths. Furthermore, we evaluate the security of the federated learning aggregation algorithm, with a specific focus on the aggregation rules and model convergence, especially in scenarios involving Byzantine adversaries.

A. Routing Performance and Security Analysis

1) Routing Performance Analysis: We assume there are k onion routers, with each onion group containing l routers. If k is not divisible by l, there will be a smaller group, but we disregard this factor for our calculations. The number of relay nodes per path is m. When messages are transmitted through the network, the success rate of message delivery is influenced by the transmission time. In our protocol, if the message transmission time exceeds the validity period of the main keys and private keys, some nodes may not be able to correctly decrypt the message, leading to transmission failure. We consider the probability of successful message delivery as a variable, let τ represent the general time, which follows a normal distribution with parameters (μ, σ), where $\tau \in (-\infty, +\infty)$; the variable t represents the actual transmission time of the message, then $t = e^{\tau}$, so $t \in (0, +\infty)$. Therefore, the probability P_T of successful message transmission between individual nodes can be calculated using the following formula:

$$P_T = \int_0^{\ln t} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\tau-\mu}{2\sigma^2}} \, d\tau. \tag{3}$$

Thus, the successful message delivery rate, P_{delivery}, from VS to VD through the node hops $\{G_1, G_2, \ldots, G_m\}$ is:

$$P_{\text{delivery}}(T) = (1 - (1 - P_T)^l)^{m+1}. \tag{4}$$

Here, l represents the size of the onion group, and $m + 1$ is the number of node hops that the message passes through during forwarding.

2) Traceability Analysis: Traceability Analysis: Traceability measures the likelihood that an attacker who has infiltrated the network and gained control over some nodes can discover the message transmission path. In paired onion routing, since the sent message includes details about the next hop, this information can be exposed if a node is compromised. For instance, if node R_2 in the path $R_1 \to R_2 \to R_3 \to R_4$ is compromised, the segment from $R_2 \to R_3$ becomes traceable. Assuming there

are k total nodes in the network, with b nodes compromised and c path segments exposed, the probability that any node is compromised is expressed as $\frac{b}{k}$. Therefore, the traceability, $P_{\text{trace}}(c)$, can be defined as follows:

$$P_{\text{trace}}(b) = \frac{1}{(m+1)^2} \sum_{i=1}^{c} \left(\frac{b}{k}\right)^{b_i} \cdot \left(1 - \frac{b}{k}\right)^{m+1-b} \cdot b_i^2. \quad (5)$$

Here, b_i represents the number of nodes on each exposed path segment, where $\sum_{i=1}^{c} b_i = b$, and $m+1$ is the total number of hops in the path.

3) Source Node Anonymity Analysis: Node anonymity refers to the ambiguity or indistinguishability of a specific node within a set of k nodes. If a source node is compromised by an attacker, its anonymity becomes zero, reducing the size of the anonymity set to 1. If the source node and the nodes in the initially contacted group G_1 remain uncompromised, the anonymity set consists of $k-b$ nodes. Collusion attacks are possible when more than two nodes are compromised, but our pairing onion routing protocol effectively mitigates such threats and is not considered further here. When nodes in G_1 that interact with the source node are compromised, the size of the source node's anonymity set, β, is:

$$\beta = n - c - g + \frac{cg}{2n}. \quad (6)$$

Let $p = \frac{b}{k}$ and $q = 1 - \frac{b}{k}$, representing all nodes in the network with ϕ. The entropy of the source node, $E_s(\phi)$, is:

$$E_s(\phi) = -q^l \sum_{i\in\phi} \tfrac{1}{k-b} \log_2\left(\tfrac{1}{k-b}\right) - (1-q^l) \sum_{i\in\phi} \tfrac{1}{\beta} \log_2\left(\tfrac{1}{\beta}\right)$$

$$= q^l \log_2(k-b) + (1-q^l)\log_2(\beta). \quad (7)$$

Therefore, the anonymity of the source node, $A_s(\phi)$, is calculated as:

$$A_s(\phi) = \frac{E_s(\phi)}{\log_2(|\phi|)}. \quad (8)$$

4) Destination Node Anonymity Analysis: This analysis is similar to the analysis of source node anonymity. If the destination node is compromised, its anonymity set shrinks to just 1. When the destination node and its adjacent group remain uncompromised, the anonymity set size is $n-c$. If nodes in the last contacted group are attacked, and considering our protocol's resistance to collusion attacks, the anonymity set size becomes g. The entropy of the destination node, $E_d(\phi)$, is calculated as follows:

$$E_d(\phi) = -q \sum_{i\in\phi} \frac{1}{k-b} \log_2\left(\frac{1}{k-b}\right) - p \sum_{i\in\phi} \frac{1}{l} \log_2\left(\frac{1}{l}\right)$$

$$= q \log_2(k-b) + p \log_2(l). \quad (9)$$

Hence, the anonymity of the destination node, $A_d(\phi)$, is:

$$A_d(\phi) = \frac{E_d(\phi)}{\log_2(|\phi|)}. \quad (10)$$

5) Path Anonymity Analysis: In anonymous routing, entropy measures the uncertainty about the origin and destination of a message, with higher anonymity increasing the difficulty for attackers to identify the actual sender and receiver. In a network with k nodes and $m+1$ hops, if b nodes are compromised and the probability of any node being attacked is $\frac{b}{k}$, the situation within each onion group (containing l nodes) is slightly different.

The probability of a node within an onion group being attacked is $\frac{bl}{k}$, while the chance of a node being selected as an onion router is $\frac{1}{l}$. Thus, the probability that a node is both chosen as a routing node and attacked is $\frac{1}{l} \cdot \frac{bl}{k} = \frac{b}{k}$, and the probability of a node being chosen as a routing node but remaining unattacked is $\left(1 - \frac{bl}{k}\right) \cdot \frac{1}{l}$. Let X denote the random variable representing the number of compromised nodes along the path. We calculate this using the binomial distribution as follows:

$$E[X] = \sum_{i=1}^{m+1} i C_{M+1}^{i} \left(\frac{b}{k}\right)^{i} \left(\frac{1}{l} - \frac{b}{k}\right)^{m+1-i}. \quad (11)$$

Assuming the attacker can guess the next hop with a probability P_{guess}, where V_i is the h-th node on the path, then:

$$P_{\text{guess}} = \begin{cases} \frac{1}{l}, & \text{if } V_i \text{ is compromised;} \\ \frac{1}{k-h}, & \text{otherwise.} \end{cases} \quad (12)$$

For simplicity, let $B = E[X]$ denote the expected number of compromised nodes on the data forwarding path. We then define the probability of successfully guessing the identity of path i as $p_i = \frac{(k-m-1+B)!}{k!} \cdot \frac{1}{l^B}$. The entropy of the path, $E_p(\varphi)$, is calculated as:

$$E_p(\varphi) = -\sum_{\forall \text{paths in } \varphi} \tfrac{(k-m-1+B)!}{k! l^B} \log_2 \tfrac{(k-m-1+B)!}{k! l^B}. \quad (13)$$

Therefore, the path anonymity $A_p(\varphi)$ can be calculated as:

$$A_p(\varphi) = \frac{E_p(\varphi)}{\log_2(|\varphi|)}. \quad (14)$$

B. Aggregation Performance and Security Analysis

1) Security Analysis: If a Byzantine adversary alters elements $d \in [1, g]$ of their vector with random values, these will markedly differ from the correct values, ensuring that outliers are detected during sampling. According to Algorithm 2, these anomalous vectors, which deviate significantly in Euclidean space from normal vectors, will be discarded.

Upon receiving model data from clients, the aggregation server selects a elements from the g-dimensional vectors of each local model. The probability that e anomalous elements are selected in the a-dimensional sampled vector is given by $\frac{C_d^e \times C_{g-d}^{a-e}}{C_g^a}$. Let the random variable Y represent the number of chosen anomalous elements. We can use the hypergeometric distribution to determine the expected value of the random variable Y as follows:

$$E[Y] = \sum_{e=0}^{d} e \frac{C_d^e \times C_{g-d}^{a-e}}{C_g^a} = a\frac{d}{g}. \quad (15)$$

This calculation indicates that on average, $\frac{a \cdot d}{g}$ anomalous elements appear in the sampled vector. Since these anomalous elements are significantly distinct from the correct elements, vectors containing such values will be excluded.

When $g - a \geq d$, the probability that a sampled vector from an adversary contains no anomalous elements is given by $\delta = \frac{C_{g-d}^{a}}{C_{g}^{a}}$; otherwise, $\delta = 0$. Let ζ be defined as the probability of removing an adversary's vector (i.e., the sampled vector contains at least one anomalous value), then:

$$\zeta = \begin{cases} 1 - \delta, & g - a \geq d \\ 1, & g - a < d \end{cases} \tag{16}$$

We focus on scenarios where $g - a \geq d$ as otherwise, all adversary vectors are automatically removed. Byzantine adversaries, with no knowledge of others' actions, randomly substitute dimensions in their vectors with anomalous values, as depicted in Fig. 5. The removal of each adversary's vector occurs as an independent event. Thus, the probability of removing k adversary vectors across f independent Bernoulli trials is calculated as $C_f^e \cdot \zeta^e \cdot (1 - \zeta)^{f-e}$. Let the random variable Z denote the number of vectors removed, and using the binomial distribution, we can determine the expected value as follows:

$$E[Z] = \sum_{e=0}^{f} e \cdot C_f^e \cdot \zeta^e \cdot (1 - \zeta)^{f-e} = f \cdot \zeta. \tag{17}$$

On average, $f \cdot \zeta$ anomalous vectors are removed per round. The vectors with the smallest scores, specifically the $Min = n - f \cdot \zeta$ vectors, are then aggregated using the coordinate median method. This minimizes the impact of adversaries on the median of each dimension when anomalous elements are randomly distributed.

Next, we discuss the range of values for the sampling rate a to ensure the correctness of the aggregation results. Under the condition $g - a \geq d$, to aggregate $n - f \cdot \zeta$ vectors, it is necessary to:

$$\begin{aligned} n - f \cdot \zeta > 2f &\implies n - \left(1 - \frac{g-a}{g}\right) f > 2f \\ &\implies \left(\frac{g-a}{g}\right) f > 3f - n \\ &\implies a < \frac{n-2f}{f} g. \end{aligned} \tag{18}$$

This means that if the sampling rate a satisfies $a < \frac{(n-2f)g}{f}$, the aggregation of $Min = n - f \cdot \zeta$ vectors will be correct because the number of anomalous elements is less than half of the set S. Even when the proportion of malicious clients is high (i.e., the number of malicious clients is close to $n/2 - 1$), AEFL can ensure the accuracy of the aggregation results by adjusting a. This conclusion helps us in selecting the appropriate sampling rate a for aggregation.

2) Convergence Analysis: In FL, essential parameters include C, the fraction of clients chosen for local training in each cycle; E, the number of training rounds each client undergoes; and B, the batch size for local training. With $C = 1$, $E = 1$, and $B = |\mathcal{D}_i|$, FL mirrors traditional distributed machine learning. In these conditions, the coordinate median aggregation method is proven to converge with an error rate of $\mathcal{Q}\left(\frac{\beta}{n\sqrt{B}} + \frac{1}{\sqrt{Bn}} + \frac{1}{B}\right)$, where β represents the fraction of Byzantine adversaries, and n denotes the total client count. To ensure the accuracy of aggregation, the size of the subset $|S|$ is maintained at greater than $2f$.

As outlined in Algorithm 2, the aggregation output is the coordinate median of subset $\mathcal{S}$ ($|\mathcal{S}| = n - f\zeta$), and Equation 17 calculates the expected number of detected anomalous vectors as $f - f\zeta$. The aggregation error rate is thus derived as $\mathcal{Q}'\left(\frac{f-f\zeta}{(n-f\zeta)\sqrt{B}} + \frac{1}{\sqrt{B(n-f\zeta)}} + \frac{1}{B}\right)$. We analyze the relationship between $\frac{f-f\zeta}{n-f\zeta} + \frac{1}{\sqrt{n-f\zeta}}$ in our coordinate median method and $\frac{f}{n} + \frac{1}{\sqrt{n}}$ in traditional median-based aggregation. Therefore, we define $F(\zeta)$ as follows:

$$F(\zeta) = \frac{f - f\zeta}{n - f\zeta} + \frac{1}{\sqrt{n - f\zeta}} - \frac{f}{n} - \frac{1}{\sqrt{n}}. \tag{19}$$

If $\zeta \in [0, 1]$, then $F(\zeta) \leq F(0) = 0$. This indicates that the convergence error rate of our AEFL aggregation scheme is lower than that of conventional median-based methods, demonstrating faster convergence.

VI. Performance Evaluation

In this section, we will evaluate the performance of the proposed framework AEFL from two aspects: routing efficiency and model aggregation efficiency through experiments.

A. Experiment Setup

1) Platforms: We employ the Simulation of Urban Mobility (SUMO) to model the movements of numerous vehicles initialized randomly on a map. The trajectories generated from this simulation are then formatted into a Network Simulator 3 (NS3) compatible version to replicate the vehicle road cooperation network. The federated learning model training was implemented in Python, while the other components were developed using C++ and the experiments were conducted on two servers running Ubuntu 18.04. These servers were equipped with two NVIDIA RTX 3090 GPUs, an Intel i9-10900K CPU, and 128GB of RAM.

2) Datasets and Clients' Models: In terms of anonymous routing, we evaluated the model's efficiency in packet transmission, node anonymity, and path reliability, comparing it with the conventional anonymous protocol TOR [36]. For federated learning aggregation, we assessed the performance of AEFL on two standard benchmark datasets. For the MNIST dataset, a CNN model was used for handwritten digit classification tasks, consisting of 60,000 training examples and 10,000 test examples, each being a 28×28 grayscale image. For the CIFAR-10 dataset, the PreResNet18 model was employed for color image classification tasks, with the dataset including 50,000 training examples and 10,000 test examples, each of size 32×32. The attack methods used in this scheme include Label Flipping Attack [37], Sybil Attack [38], Scaling Attack

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3446725

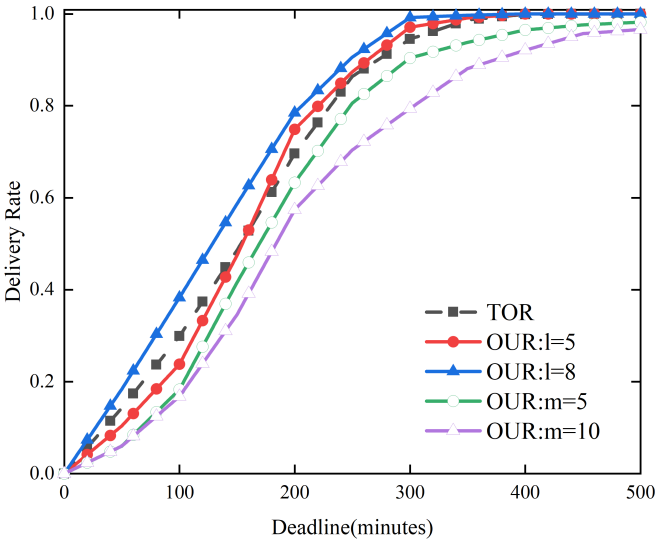

Fig. 6. Comparison of packet delivery rates under different conditions.

[39], LIE Attack [40], and Gaussian Attack [41]. We evaluated the detection accuracy of malicious clients and the training accuracy of the global model.

3) Metrics and the Baseline: In this experiment, we set the size of the vehicle road cooperation network to 300 nodes and grouped them accordingly. We configured the number of participating federated learning vehicle clients to 50, with 10 of them being Byzantine adversaries, and set the number of communication rounds to 200. During the experiments, we used the same model structure, with a batch size $B = 128$ and a learning rate $\eta = 0.001$. The local training rounds for both datasets were set to $E = 1$. Finally, the training dataset was divided into multiple subsets to be allocated to the clients.

B. Experiment Results

1) Group Pairing Onion Routing Efficiency Evaluation: Fig. 6 presents a comparison of the packet delivery rates between the TOR and AEFL across different variables. The packet delivery rate is determined by the proportion of packets successfully received relative to the total number of packets sent across the network. The findings demonstrate that at the same time interval, when the number of relay nodes is $m = 3$, the success rate of message delivery for a group with $l = 8$ nodes is higher compared to a group with $l = 5$ nodes. Additionally, for groups of size $l = 5$, the delivery rate decreases as the number of relay nodes increases. Our scheme, in comparison to TOR, achieves a 100% delivery rate in a shorter timeframe. This improvement is due to an increase in the number of nodes per group, which enhances the probability of successful forwarding, thereby leading to a higher delivery rate.

Fig. 7 illustrates the comparison of node anonymity between AEFL and TOR under scenarios where different numbers of onion routers are compromised. Fig. 7(a) focuses on source node anonymity. The experimental results indicate that, when the group size is fixed, an increase in the number of infected nodes leads to a reduction in source node anonymity. This

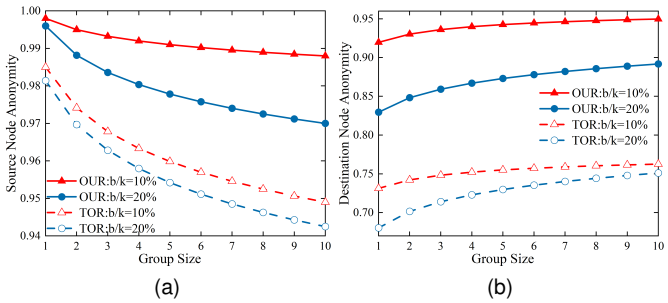

Fig. 7. Comparison of (a) source node anonymity and (b) destination node anonymity with different numbers of compromised nodes.

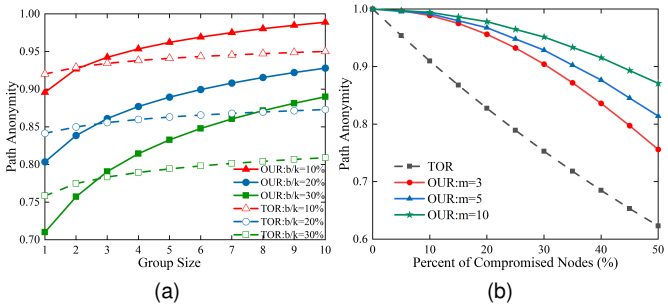

Fig. 8. Comparison of path anonymity when (a) the number of relay nodes $m = 3$ and (b) the group size $l = 5$.

decrease is attributed to the increased probability that the first group of nodes in contact with the source node will be compromised as the number of infected nodes rises. Our approach significantly surpasses the TOR scheme in preserving source node anonymity. Fig. 7(b) addresses destination node anonymity. The results show that when the group size remains constant and the number of infected nodes increases, the anonymity of the destination node improves. This enhancement occurs because an increase in the number of nodes within the group decreases the likelihood of compromising nodes along the information transmission path, thus increasing destination node anonymity. When 10% of the nodes are under attack and the group size is 10, our scheme achieves a destination node anonymity of approximately 0.95, whereas the TOR scheme only reaches about 0.76 under the same conditions.

Fig. 8 demonstrates a comparison of path anonymity between AEFL and TOR under various parameter settings. Fig. 8(a) analyzes path anonymity in relation to group size versus the proportion of infected nodes. The results indicate that with a fixed number of relay nodes ($m = 3$), path anonymity increases with an increase in group size. This is attributed to the augmentation in the number of groups, which decreases the likelihood of compromising intermediate nodes in routing paths, thereby enhancing path anonymity. When the proportion of attacked nodes is 10% and the group size is 10, our scheme achieves a destination node anonymity of approximately 0.99, in contrast to the TOR scheme, which only achieves about 0.92 under the same conditions. Fig. 8(b) evaluates path anonymity as it relates to the proportion of infected nodes versus the number of relay nodes. The findings demonstrate that for a

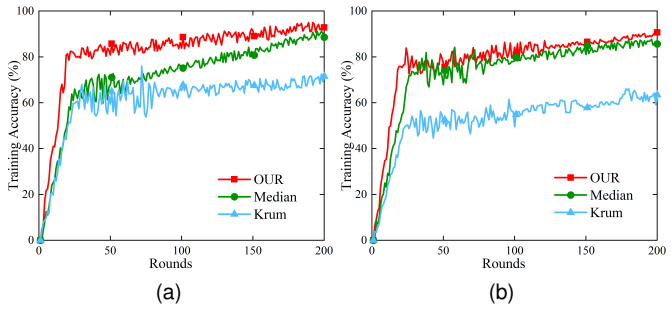

Fig. 9. Comparison of training accuracy of different federated learning aggregation algorithms on the (a) MNIST dataset and (b) CIFAR-10 dataset.

TABLE I
DETECTION RESULTS OF MALICIOUS CLIENTS UNDER DIFFERENT ATTACKS.

Attack Method	MNIST	CIFAR-10
Label Flipping Attack	0.98	0.96
Sybil Attack	0.99	0.99
Scaling Attack	0.97	0.93
LIE Attack	0.94	0.98
Gaussian Attack	0.95	0.92

group size of $l = 5$, the more routing relay nodes there are, the higher the path anonymity. However, as the number of compromised nodes increases, path anonymity decreases even with the same number of relay nodes. The highest path anonymity, consistently maintained above 0.87, occurs with 10 relay nodes. When the number of infected nodes is consistent, our scheme shows superior anonymity compared to TOR.

2) Federated Learning Aggregation Efficiency Evaluation: Table I compares the detection accuracy of malicious clients under different attacks for both datasets in AEFL. Detection accuracy is defined as the proportion of all correct predictions out of the total sample size (including both malicious and honest clients). The results indicate that the model exhibits robust performance against Label Flipping, Sybil, Scaling , LIE and Gaussian Attacks. Notably, in Sybil Attacks, the detection accuracy for Byzantine adversaries in both datasets can reach up to 99%. This high accuracy is attributed to the consistent attack behaviors of malicious clients, making them more easily identifiable by the sampling algorithms.

Fig. 9 presents the accuracy of federated learning model training of AEFL, and we have designed comparative experiments to contrast our scheme with two classic Byzantine robust aggregation schemes: Median and Krum. Fig. 9(a) shows the model accuracy under the MNIST dataset, while Fig. 9(b) represents the model accuracy under the CIFAR-10 dataset. The results indicate that our scheme exhibits good Byzantine robustness, with model convergence performance close to that of the Median algorithm. Overall, our scheme demonstrates the best model convergence results, whereas Krum exhibits less stable convergence.

VII. CONCLUSION

In this paper, addressing the issues of user privacy parameter leakage and Byzantine attacks within vehicle road cooperation networks that have implemented AIoT, we have designed an anonymous and efficient federated learning system framework, termed AEFL. Utilizing onion routing technology, we introduced a group pairing onion routing protocol, which secures the communication process by encrypting local data packets layer by layer and decrypting them sequentially along the transmission path. Moreover, we implemented a sampling-based Byzantine node detection algorithm, ensuring the security and efficiency of the federated learning model aggregation process. Theoretical analysis of AEFL's performance and security was conducted, and experimental results indicate that our framework effectively achieves data anonymity and federated learning aggregation efficiency.

In future work, we will focus on reducing the communication overhead in vehicle road cooperation networks implementing AIoT and ensuring the security of private data within the aggregation server.

REFERENCES

[1] M. Mohammadi and A. Al-Fuqaha, "Enabling cognitive smart cities using big data and machine learning: Approaches and challenges," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 94–101, 2018.

[2] R. Xu, X. Xia, J. Li, H. Li, S. Zhang, Z. Tu, Z. Meng, H. Xiang, X. Dong, R. Song *et al.*, "V2v4real: A real-world large-scale dataset for vehicle-to-vehicle cooperative perception," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 712–13 722.

[3] P. Rani and R. Sharma, "Intelligent transportation system for internet of vehicles based vehicular networks for smart cities," *Computers and Electrical Engineering*, vol. 105, p. 108543, 2023.

[4] J. Zhang and D. Tao, "Empowering things with intelligence: a survey of the progress, challenges, and opportunities in artificial intelligence of things," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 7789–7817, 2020.

[5] F. Samie, L. Bauer, and J. Henkel, "From cloud down to things: An overview of machine learning in internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4921–4934, 2019.

[6] 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.

[7] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE signal processing magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[8] Z. Yu, J. Hu, G. Min, Z. Zhao, W. Miao, and M. S. Hossain, "Mobility-aware proactive edge caching for connected vehicles using federated learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5341–5351, 2020.

[9] J. Mills, J. Hu, and G. Min, "Multi-task federated learning for personalised deep neural networks in edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 3, pp. 630–641, 2021.

[10] J. Wang, J. Hu, J. Mills, G. Min, M. Xia, and N. Georgalas, "Federated ensemble model-based reinforcement learning in edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 6, pp. 1848–1859, 2023.

[11] J. Mills, J. Hu, and G. Min, "Communication-efficient federated learning for wireless edge intelligence in iot," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5986–5994, 2019.

[12] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 691–706.

[13] X. Zhang, X. Chen, M. Hong, Z. S. Wu, and J. Yi, "Understanding clipping for federated learning: Convergence and client-level differential privacy," in *International Conference on Machine Learning, ICML 2022*, 2022.

[14] H. Goyal and S. Saha, "Multi-party computation in iot for privacy-preservation," in *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2022, pp. 1280–1281.

[15] H. Mahdikhani, R. Lu, J. Shao, and A. Ghorbani, "Using reduced paths to achieve efficient privacy-preserving range query in fog-based iot," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4762–4774, 2020.

[16] R. Dingledine, N. Mathewson, P. F. Syverson *et al.*, "Tor: The second-generation onion router." in *USENIX security symposium*, vol. 4, 2004, pp. 303–320.

[17] R. W. Lai, K.-F. Cheung, S. S. Chow, and A. M.-C. So, "Another look at anonymous communication," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 5, pp. 731–742, 2018.

[18] V. Shejwalkar and A. Houmansadr, "Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning," in *NDSS*, 2021.

[19] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*. Pmlr, 2018, pp. 5650–5659.

[20] D. Alistarh, Z. Allen-Zhu, and J. Li, "Byzantine stochastic gradient descent," *Advances in neural information processing systems*, vol. 31, 2018.

[21] H. Yang, X. Zhang, M. Fang, and J. Liu, "Byzantine-resilient stochastic gradient descent for distributed learning: A lipschitz-inspired coordinate-wise median approach," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 5832–5837.

[22] 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.

[23] L. Lyu, J. Yu, K. Nandakumar, Y. Li, X. Ma, J. Jin, H. Yu, and K. S. Ng, "Towards fair and privacy-preserving federated deep models," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 11, pp. 2524–2541, 2020.

[24] A. M. Girgis, D. Data, S. Diggavi, P. Kairouz, and A. T. Suresh, "Shuffled model of federated learning: Privacy, accuracy and communication trade-offs," *IEEE journal on selected areas in information theory*, vol. 2, no. 1, pp. 464–478, 2021.

[25] Y. Li, X. Tao, X. Zhang, J. Liu, and J. Xu, "Privacy-preserved federated learning for autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 8423–8434, 2021.

[26] J. Domingo-Ferrer, A. Blanco-Justicia, J. Manjón, and D. Sánchez, "Secure and privacy-preserving federated learning via co-utility," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3988–4000, 2021.

[27] Y. Chen, Y. Su, M. Zhang, H. Chai, Y. Wei, and S. Yu, "Fedtor: An anonymous framework of federated learning in internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 18 620–18 631, 2022.

[28] D. Catalano, D. Fiore, and R. Gennaro, "A certificateless approach to onion routing," *International Journal of Information Security*, vol. 16, pp. 327–343, 2017.

[29] P. Kotzanikolaou, G. Chatzisofroniou, and M. Burmester, "Broadcast anonymous routing (bar): scalable real-time anonymous communication," *International Journal of Information Security*, vol. 16, pp. 313–326, 2017.

[30] G. Han, L. Zhou, H. Wang, W. Zhang, and S. Chan, "A source location protection protocol based on dynamic routing in wsns for the social internet of things," *Future Generation Computer Systems*, vol. 82, pp. 689–697, 2018.

[31] S. Kim, J. Han, J. Ha, T. Kim, and D. Han, "Sgx-tor: A secure and practical tor anonymity network with sgx enclaves," *IEEE/ACM Transactions on Networking*, vol. 26, no. 5, pp. 2174–2187, 2018.

[32] R. Guerraoui, S. Rouault *et al.*, "The hidden vulnerability of distributed learning in byzantium," in *International Conference on Machine Learning*. PMLR, 2018, pp. 3521–3530.

[33] D. Cao, S. Chang, Z. Lin, G. Liu, and D. Sun, "Understanding distributed poisoning attack in federated learning," in *2019 IEEE 25th international conference on parallel and distributed systems (ICPADS)*. IEEE, 2019, pp. 233–239.

[34] L. Muñoz-González, K. T. Co, and E. C. Lupu, "Byzantine-robust federated machine learning through adaptive model averaging," *arXiv preprint arXiv:1909.05125*, 2019.

[35] H. Hu, R. Peng, Y.-W. Tai, and C.-K. Tang, "Network trimming: A data-driven neuron pruning approach towards efficient deep architectures," *arXiv preprint arXiv:1607.03250*, 2016.

[36] K. Sakai, M.-T. Sun, W.-S. Ku, J. Wu, and F. S. Alanazi, "Performance and security analyses of onion-based anonymous routing for delay tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 12, pp. 3473–3487, 2017.

[37] X. Cao, M. Fang, J. Liu, and N. Z. Gong, "Fltrust: Byzantine-robust federated learning via trust bootstrapping," *arXiv preprint arXiv:2012.13995*, 2020.

[38] C. Fung, C. J. Yoon, and I. Beschastnikh, "The limitations of federated learning in sybil settings," in *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*, 2020, pp. 301–316.

[39] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International conference on artificial intelligence and statistics*. PMLR, 2020, pp. 2938–2948.

[40] G. Baruch, M. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[41] X. MA, Q. LI, Q. JIANG, Z. MA, S. GAO, Y. TIAN, and J. MA, "Byzantine-robust federated learning over non-iid data," *Journal on Communications*, vol. 44, no. 6, pp. 138–153, 2023.