Secure Aggregation for Privacy-preserving Federated Learning in Vehicular Networks

SANGHYUN BYUN, University of Colorado Colorado Springs, Colorado Springs, United States
ARIJET SARKER, University of Colorado Colorado Springs, Colorado Springs, United States
SANG-YOON CHANG, University of Colorado Colorado Springs, Colorado Springs, United States
JUGAL KALITA, University of Colorado Colorado Springs, Colorado Springs, United States

The automotive industry has been enhancing autonomous driving systems utilizing the computation and communication networks embedded in vehicles (e.g., cellular networks and sensors) and roadside units (e.g., radar and cameras). Robust security and privacy requirements are essential in Intelligent Transportation Systems (ITSs). To satisfy these requirements, most developed autonomous driving systems (e.g., Waymo and Tesla) use machine learning. Machine learning models trained on sensitive raw data promise improvements in performance; however, they cannot provide privacy for sensitive raw data and users. Federated learning advances privacy-preserving distributed machine learning by aggregating the model parameter updates from individual devices in a secure manner. Security Credential Management System (SCMS) for Vehicle to Everything (V2X) communication provides a guarantee for authentication in a privacy-preserving manner and punishes misbehaving vehicles through misbehavior reporting. In this article, we design a secure aggregation protocol for privacy-preserving federated learning for vehicular networks. Our protocol allows a server to verify vehicles in a secure manner and is used to aggregate each vehicle-provided global model update for federated learning. We prove our protocol for security in the honest but curious framework and detect active adversary attacks, as well as show that it provides trust in different domains (e.g., SCMS and outside the domain of SCMS) and in a privacy-preserving manner for vehicles using SCMS. We analyze the process of federated learning in each vehicle and server while communicating during driving on several types of roads (e.g., local, urban, and rural) using cellular networks (LTE and 5G).

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

Additional Key Words and Phrases: Public key infrastructure, security credential management system, federated learning, vehicular networking, cellular network

ACM Reference Format:
Sanghyun Byun, Arijet Sarker, Sang-Yoon Chang, and Jugal Kalita. 2024. Secure Aggregation for Privacy-preserving Federated Learning in Vehicular Networks. *ACM J. Auton. Transport. Syst.* 1, 3, Article 14 (July 2024), 25 pages. https://doi.org/10.1145/3657644

Authors' Contact Information: Sanghyun Byun, Engineering and Applied Science, University of Colorado Colorado Springs, Colorado Springs, Colorado, United States; e-mail: sbyun@uccs.edu; Arijet Sarker, Engineering and Applied Science, University of Colorado Colorado Springs, Colorado Springs, Colorado, United States; e-mail: asarker@uccs.edu; Sang-Yoon Chang, Engineering and Applied Science, University of Colorado Colorado Springs, Colorado Springs, Colorado, United States; e-mail: schang2@uccs.edu; and Jugal Kalita, Engineering and Applied Science, University of Colorado Colorado Springs, Colorado Springs, Colorado, United States; e-mail: jkalita@uccs.edu.

ACM 2833-0528/2024/07-ART14
https://doi.org/10.1145/3657644

1 INTRODUCTION

During the Fourth Industrial Revolution, automotive industries have enhanced self-driving systems, leveraging the computation and communication network embedded in vehicles using machine learning [20, 23]. However, their self-driving models do not provide privacy for users and data while collecting datasets about the driving experience. **Federated learning (FL)** with a secure aggregation protocol [5] has been proposed as a distributed machine learning mechanism for protecting the privacy of data and users. The **3rd Generation Partnership Project (3GPP)** has been developing standards to meet critical vehicular communication requirements such as low latency, high speed, and reliability by introducing **Ultra-Reliable Low Latency Communication (URLLC)** [1, 26]. The backbone of a vehicular network is the on-board intelligence capabilities in vehicles to process real-time data (BSMs, e.g., sender's time, position, and speed; traffic flow, road, and network conditions) collected through sensors such as cameras, RADAR, LIDAR, and mechanical control units. Thus, it is critical to protect real datasets of driving experiences to improve the self-driving system against adversarial attacks while maintaining the privacy of the vehicles. In this regard, the **US Department of Transportation (USDOT)** has proposed a **Public Key Infrastructure (PKI)** called the **Security Credential Management System (SCMS)** [7] for preserving vehicular privacy and reliable communication by issuing pseudonym certificates to vehicles. The generation and provisioning of those certificates are divided among multiple organizations to achieve the privacy requirements of the vehicles.

SCMS has the capacity to maintain vehicular privacy and provide two to several orders of magnitude larger numbers of certificates (300 billion certificates per year) than the current largest existing PKIs, e.g., Europay-MasterCard Visa Consortium [7] and Defense Information Systems Agency for the Common Access Cards program [30]. Also, SCMS is a consortium project among the USDOT, **National Highway Traffic Safety Administration (NHTSA)**, and other top commercial automobile companies, e.g., Honda, Ford, General Motors, Nissan, Mazda, and Volkswagen. These factors motivate us to choose SCMS. Though SCMS supports a range of applications, e.g., communicating basic safety messages, enabling vehicle turning, and maintaining an intelligent traffic signal system [27] to improve the driving experience and large-scale PKI communication for vehicles, vehicles may still need to communicate with additional applications and services that are outside of SCMS-controlled applications (controlled by a different authority). Since the certificate of the servers (hosting these external applications and services) is not provided by SCMS, the following research questions may arise: (1) how the vehicle can trust that it is communicating with the authorized external applications and services and (2) how applications and services can trust that they are communicating with the authorized vehicles.

FL is an emerging distributed machine learning paradigm using a network environment. In particular, individual clients collect their own raw data as they deploy existing machine learning models and then try to train from raw data in individual devices, meaning that they do not need to share private data. In the FL setting, a central server repeatedly tried to communicate with clients to perform the global model updates without the need to share their private data. The server aggregates the individual model updates to generate the global model, which is broadcast to all clients. FL takes into account the mutually untrusted relationships between clients and servers and may be vulnerable to poisoning attacks, where malicious clients may try to manipulate the global model update, producing a negative effect on the model accuracy. In FL research, there is a notable emphasis on developing Byzantine-robust aggregation FL algorithms. However, it does not provide robustness to protect against poisoning attacks if the number of clients increases in the FL process. Therefore, this article describes the design of the FL protocol in vehicular networks. The interface specification of the current version of our proposed article is available from [9]. This

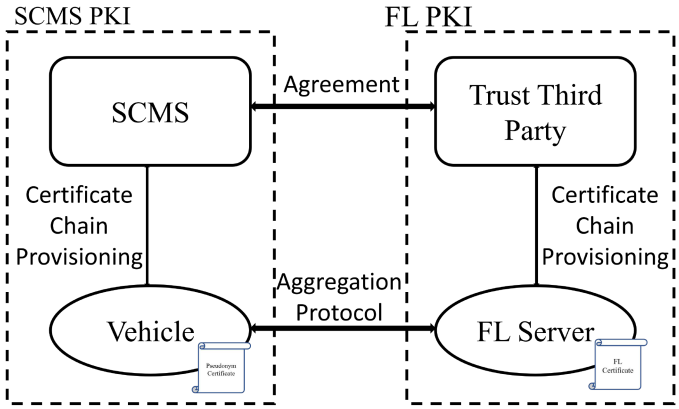


Fig. 1. High-level overview of proposed architecture between SCMS and FL PKI.

article introduces adding a special revocation process between SCMS and FL PKI as well as an aggregation protocol for FL that contains several concepts including misbehavior detection for FL and recovery of poisoning global model updates. We focus on the aggregation protocol utilizing the privacy aspect of SCMS to preserve vehicular privacy during the FL process between vehicles and the **Federated Learning Server (FLS)**. We consider the FL application (used to enable data-driven machine learning for better self-driving experience while protecting data privacy) as outside of an SCMS-controlled application. In FL, FLS collects and trains data from vehicles in a distributed manner. Though FL is designed in such a manner that entities without users are unable to link raw data, it is still necessary to maintain users' privacy, specifically location privacy in vehicular networks, thus protecting against the location tracking of the vehicles by the attacker. Figure 1 provides a high-level overview of the proposed communication structure between SCMS and FL PKI. We consider the need for an FL secure aggregation protocol on the vehicular network that

— builds trust relationship between SCMS and FL PKI by sharing only the minimum required identity data,
— provides preserving privacy for the user and data,
— detects malicious activities for FL and is robust to the user dropping out, and
— measures the real-world communication latency during driving between the users of the respective PKIs (SCMS and FL PKI).

The rest of the article is organized as follows. Section 2 provides background and related works on SCMS and FL. We present a problem statement establishing the scope of the contribution and the threat model in Section 3. Section 4 describes the design principle of our proposed approach. Section 5 explains the misbehavior detection and revocation process in our work. In Section 6, the security analysis is explained. The evaluation details are described in Section 7. This is followed by the discussion and conclusion in Sections 8 and 9.

2 BACKGROUND AND RELATED WORKS

We provide a description of SCMS in Section 2.1, FL in Section 2.2, secure aggregation protocol in Section 2.3, and robust aggregation algorithms in Section 2.4.

2.1 Security Credential Management System

SCMS is a PKI for vehicles, as well as infrastructure for **vehicle-to-vehicle (V2V)** and **vehicle-to-everything (V2X)** communication to maintain vehicular privacy by issuing digital

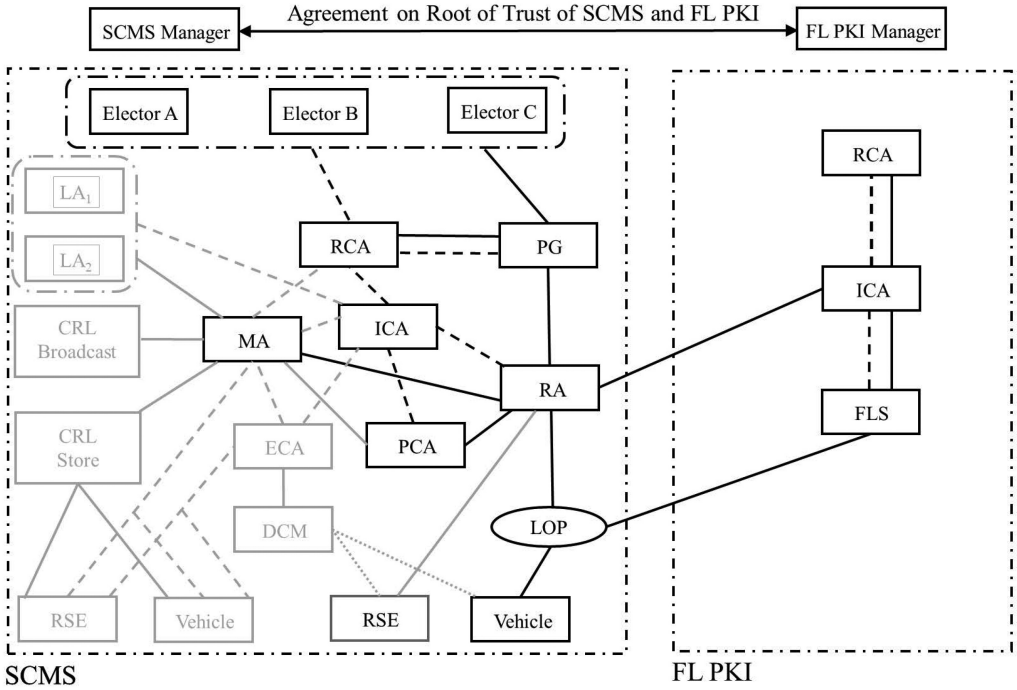


Fig. 2. Proposed structure between SCMS PKI and FL PKI.

certificates [7, 29]. The threat model of SCMS includes attackers that are SCMS insiders involved in the digital certificate production as well as SCMS outsiders capable of accessing the V2X channels during certificate use [8, 14]. Such a threat model is motivated by the generation and provisioning of certificates among multiple authorities in SCMS, thus incorporating a distributed approach so that a single point of compromise cannot breach vehicular privacy. The dotted rectangular box on the left of Figure 2 depicts the authorities and their interactions (using the lines) involved in SCMS. The dotted line shows the credential chain of trust in SCMS, meaning one authority provides a certificate to the other, and the solid line shows SCMS communication, meaning one component can send information to the other. We intentionally blur a few authorities and lines in SCMS because they are outside the scope of this article. Note that SCMS provides security mechanisms to protect against several attacks, such as Man-in-the-Middle attacks, and unauthorized access.

The SCMS Manager, serving as a centrally pivotal component with exactly one instance in the system instantiation, formulates organizational and technological policies. It establishes methodologies for scrutinizing misbehavior and evaluating revocation requests to ensure correctness and fairness. Electors, positioned as the focal points of trust in SCMS, engage in a distributed voting process to decide on the addition or revocation of a **Root Certificate Authority (RCA)** or an elector within SCMS. The initial set of electors is implicitly trusted. The RCA, denoted as RCA_{SCMS}, holds a self-signed certificate endorsed by the electors. This RCA issues certificates to both the **Intermediate Certificate Authority (ICA)** and the **Policy Generator (PG)**. The ICA, safeguarding the RCA from potential threats and attacks, issues certificates to the **Registration Authority (RA)** and the **Pseudonym Certificate Authority (PCA)**. The PG is responsible for signing and maintaining a **Global Certificate Chain File (GCCF)** containing all trusted certificate chains within SCMS. The RA manages the verification and processing of requests for generating pseudonym

certificates from vehicles, specifically the **onboard equipment (OBE)** in a vehicle. It provides pseudonym certificates to vehicles, while the PCA is responsible for the actual generation of pseudonym certificates. The RA also generates a **Local Certificate Chain File (LCCF)** from the GCCF and obtains its signature from the PG. The LCCF stores the certificate chain of all PCAs and certificates from other authorities necessary for interaction or trust by the vehicles. Note that PCA knows the content of the pseudonym certificate without knowledge of the entity for which it is generated. Simultaneously, the RA possesses information about the entity requesting the pseudonym certificate without knowledge of the certificate's content during provisioning. The **Location Obscure Proxy (LOP)** plays a role in concealing the location of the requesting vehicle within the SCMS architecture.

2.2 Federated Learning

FL has been proposed as a distributed machine learning mechanism with the aim of protecting the data privacy of end users, e.g., vehicles. In FL, a central server, e.g., a training server or service provider (which we refer to as FLS in this article), initializes the global machine learning model and sends the model to the end users. The end users deploy the global model, generate local learned models that are trained by collecting raw data locally, and send the learned models to the central server using a secure aggregation protocol. Finally, the central server generates the new global model by aggregating trained local models received from the end users according to the secure aggregation protocol. The processes in FL are repeated in multiple rounds until the global model attains the required performance. In particular, FL focuses on preserving the privacy of raw data in local devices, which means that the trained local models (sent by end users) do not include raw data because of enhanced privacy regulations or laws such as the **General Data Protection Regulation (GDPR)** in the EU, the National Security Law in China, and the federal **Personal Information Protection and Electronic Documents Act (PIPEDA)** in Canada. The central server cannot access the local raw data or involve itself in the processing of the local models of the end users. However, the FL process can be compromised at various points such as the optimization algorithm (e.g., **stochastic gradient descent (SGD)**), the aggregation rules, the end user's raw data, the network, and the server for machine learning [6].

2.3 Secure Aggregation Protocol

The FL secure aggregation protocol [5] utilizes a PKI for authorization. Clients sign messages with their identities, and others, including clients and the server, can verify these signatures through registered identities, typically represented by key pairs. PKI prevents the server from posing as any arbitrary client in the network. The protocol's key exchange process involves advertising, sharing, and collecting masked inputs. Consistency checks are crucial, encompassing the signing and broadcasting of public keys, validation steps, and the generation of secret shares. Subsequent rounds feature unmasking, where surviving users contribute to the final aggregated value. The protocol is engineered to ensure security against an active adversary, incorporating multiple rounds with specific actions and validations. A notable aspect is the privacy-preserving element achieved through group share key fairs computed from Secret Sharing. During communication between clients and the server, they employ group share key pairs to maintain privacy while retaining ownership of their identities through key pairs.

2.4 Existing Robust Aggregation Algorithm for FL

In this section, we describe existing robust aggregation algorithms for FL [2–4, 13, 22, 31, 32]. We do not consider the federated average because it is not a robust aggregation algorithm.

— The **Adaptive Federated Average algorithm [22, 31]** tries to discard malicious global model updates based on their cosine similarities through a benign gradient. In particular, it computes a weighted average of collected model updates and cosine similarities to the weighted average of collected model updates. Then, it removes the malicious global model updates that are not within the expected similarity range.
— The **Krum algorithm [4, 13]** leverages the insight that malicious gradients must be sufficiently distant from benign gradients to effectively poison the global model. It selects a single model update among the client's input as the aggregation model update in each iteration. From the aggregation model update for each client, we use Krum to compute a score that is the sum of the distance of I - m - 2 nearest neighbors of input, where I is the total number of clients and m is the number of misbehaving clients.
— The **Multi-Krum algorithm [4, 13]** has been modified to Multi-Krum to enhance the utilization of shared knowledge among clients in each FL round. In this adaptation, Multi-Krum selects gradients using the Krum method from a remain, adds them to a selection, and removes them from the remain. This iterative process continues until g gradients are selected, where I - $g > 2m + 2$. Finally, the algorithm averages the gradients in the selection. Multi-Krum consistently outperforms Krum in terms of global model accuracy, showcasing its efficacy in leveraging client knowledge.
— **Bulyan [2, 13]** highlights the vulnerability where a malicious gradient can strategically maintain proximity to benign gradients by having a single gradient dimension with an exceptionally large value on the order of $\Omega(\sqrt{p}/d))$, thereby impeding the convergence of the global model. In response to this challenge, the effectiveness of the proposed solution for robustness is ensured when the requirement $I \geq 4m + 3$. The initial step involves selecting θ gradients, where θ is constrained by $\theta \leq n - 2m$, using a method akin to Multi-Krum. Subsequently, Bulyan computes the Trimmed-mean of the selected gradients to further refine the aggregation process.
— The**Trimmed-mean algorithm [3, 32]** aggregates each dimension of input gradients separately. Specifically, for a given dimension j, it arranges the values of the j th dimension across all gradients, denoted as $\triangledown\{j \in [I]\}$. Subsequently, it sorts these values, removes the β largest and smallest values, and computes the average of the remaining values as the aggregate for dimension j. In the context of Trimmed-mean, the parameter β is set equal to the number of malicious clients. It demonstrates order-optimal error rates when m is within the range of $\beta \geq m \geq i/2$, particularly for strongly convex objective functions. This demonstrates the effectiveness of Trimmed-mean in achieving optimal performance under certain conditions.

3 PROBLEM STATEMENT

3.1 Threat Model

We follow the threat model with the risk assessment that there are risks to privacy from security attacks by SCMS insiders as well as outsiders and poisoning attacks for FL. We specifically discuss in detail the attacker's goals, capabilities, and attacks based on knowledge.

— **Attacker's goals:** In FL in SCMS, attackers try to increase the test error rate of the global model for a large number of inputs for poison prediction of the global model due to an untargeted poisoning attack and a targeted poisoning attack.
— **Attacker's capabilities:** We assume that the adversaries cannot compromise the authorities involved in the PKI systems and break the cryptographic primitives, i.e., digital signatures and hash collisions. However, the attacker can monitor the communication between the vehicles and the FLS and identify the vehicles if privacy mechanisms (i.e., providing pseudonym

certificates) are not preserved. The malicious vehicle may have its privacy compromised from SCMS insiders or outsiders like impostors that inject themselves into the FL system between vehicles and the FLS.

— **Attacker's attacks based on knowledge:** If the attack is based on partial knowledge, attackers gain knowledge of the global model, the loss function, local training data, and model updates on malicious vehicles. An attack based on full knowledge assumes that the attacker knows model updates, local training data, and aggregation algorithms on all vehicles. For instance, the adversary possesses knowledge about both the gradients of benign devices and the server's aggregation algorithm. It allows the server to evaluate the robustness of its aggregation algorithms. On the other hand, the adversary lacks access to the gradients of benign devices. To compute malicious gradients, this adversary utilizes benign gradients calculated using benign data on devices under its control.

3.2 Design Goals

We aim to design an FL platform for vehicular networks to preserve the privacy of both vehicle and data for the FL process, as well as to detect adversarial attacks and recover from poisoning of global model updates for FL using published detection and recovery methods. Our platform should allow an FL process only between the authorized end-entities (vehicle and FLS) while preserving the privacy of the vehicle. Specifically, our design goals are as below:

— **Privacy:** We use the pseudonym certificate for vehicles in SCMS. PCA creates a pseudonym certificate through the linkage value to be signed by PCA and then it reconstructs a new private key for each pseudonym certificate. Therefore, we provide privacy of the vehicle from the pseudonym certificate provisioning process [7].

— **Independent of SCMS and FL PKI:** SCMS does not provide a certificate chain to all vehicular services if services are necessary. It needs to manage trustworthy relationships between SCMS and entities outside of the SCMS and protect each entity from being compromised. Thus, we aim to design an independent FL platform. In particular, it must not have effects on both SCMS and FL PKI if the entities of SCMS and FL PKI are compromised. The complexity of the management of SCMS increases if SCMS absorbs the FL PKI.

— **Independent of robust aggregation algorithms, detection methods, and recovery methods:** Various robust aggregation algorithms have been proposed to prevent the poisoning of global model updates for use during training using aggregation algorithms. We aim to design a protocol for vehicles that is compatible with any aggregation algorithm. Detection and recovery methods also have been proposed to detect malicious vehicles for FL and to recover poisoning global model updates from malicious vehicles. Moreover, both detection and recovery methods may be developed in the future. Therefore, our goal is to develop an FL platform for vehicular networks, with a focus on achieving superior performance.

4 DESIGN PRINCIPLE

In this section, we discuss the components of SCMS and FL PKI, specifying their roles related to our proposed structure. Most importantly, we explain the certificate management—root agreement, certificate chain validation, and provisioning process of the two PKIs, SCMS and FL PKI. We also discuss our secure aggregation protocol based on the certificate management between SCMS and FL PKI. Table 1 shows the acronyms used in our approach.

4.1 Proposed Structure

The components of SCMS and FL PKI with their roles in our proposed communication structure are described in this section. Figure 2 depicts the components of the proposed communication

Table 1. Acronyms

Acronym	Explanation
SCMS	Security Credential Management System
FLS	Federated Learning Server
FL PKI	Federated Learning Public Key Infrastructure
RCA	Root Certificate Authority
ICA	Intermediate Certificate Authority
PG	Policy Generator
PCA	Pseudonym Certificate Authority
RA	Registration Authority
LOP	Location Obscure Proxy
MA	Misbehavior Authority
CRL	Certificate Revocation List
FLS	Federated Learning Server
GCCF	Global Certificate Chain File
LCCF	Local Certificate Chain File
CCF	Certificate Chain File

interactions and their logical roles. We explain the logical interaction roles between the SCMS and FL PKI, excluding the logical roles of components inside their original PKI. There are three types of connections in our proposed communication structure:

— The solid line with arrows at both ends represents the unique secure communication channel between SCMS Manager and FL PKI Manager.
— The dotted lines show the credential chain of trust that is used to provide a certificate by one component to the other and verify the certificate by the respective components.
— The solid lines show the secure communication that indicates sending information by one component to the other.

The components considered in SCMS and FL PKI for our proposed approach communicate with each other using a secure communication channel and protocols, i.e., **Transport Layer Security (TLS)** suite [15]. Data encryption and authentication happen at the application layer if the data is forwarded via a component that is not designed to access it. For example, the RA cannot decrypt the data sent by the ICA of FL that is addressed to the electors. We discuss the core functionalities of the SCMS components related to our work in Section 2.1, whereas this section includes the description of additional functionalities of the SCMS components to maintain the communication with FL PKI as well as the functionalities of FL PKI components. For this purpose, the following components in SCMS are used in our proposed structure:

— SCMS Manager: The SCMS Manager ensures efficient and fair operation between the SCMS and FL PKI, which specify the organizational and technical policies of SCMS and FL PKI. This includes designing the policies for agreement on the root of trust, endorsing and revoking the certificate chain from FL PKI. The SCMS Manager constructs the policies for handling misbehavior requests from FL Manager.
— Elector: Electors sign ballots to endorse or revoke a certificate chain of FL PKI by the agreement on the root of trust between SCMS and FL PKI. Those ballots with the least quorum votes of the electors establish a trust relationship between SCMS and FL PKI.
— RCA: The RCA of SCMS signs the certificate chain of FL PKI given the condition that the certificate chain of FL PKI gets the majority vote from electors. The Certificate Chain of FL PKI signed by RCA of SCMS and a ballot with a quorum vote of electors establishes trust

between SCMS and FL PKI. An entity related to the FL PKI verifies any certificate by chain file at hand to the trust RCA system in SCMS. As the fundamental concept of any PKI, if RCA's private key is broken, the system is potentially compromised.

— PG: PG manages the **Global Policy File (GPF)**, which involves configuration information of FL PKI and the GCCF of FL PKI, containing all of the trust chain of SCMS including the certificate chain file of FL PKI. It also handles the voting process of electors and sends the voting result to RCA of SCMS.
— **Misbehavior Authority (MA)**: MA processes reports of misbehavior or malfunctioning by vehicles from FLS, and then it revokes, adds them to **Certificate Revocation List (CRL)**, and sends CRL to FLS.
— ICA: ICA of SCMS provides certificates to RA and PCA while ICA of FL PKI provides certificates to FLS. ICA acts like a secondary gatekeeper to protect the root CA from traffic and attacks.
— RA: RA is the first gatekeeper for validation from FLS, which receives requests for authenticated information about SCMS and FL PKI configuration changes to SCMS or FL PKI, containing entities networking information, certificate, or relaying policy issued by SCMS or FL PKI manager. RA also creates an LCCF, a subset of GCCF. In our case, LCCF contains all the certificate chain files needed for communication with FL PKI.
— Vehicle: Vehicle is the end-entity in SCMS to use the **Pseudonym Certificate (PC)** to authenticate and communicate with FLS.

4.2 Certificate Management

In this subsection, we describe the process of agreement between SCMS PKI and FL PKI. We also explain the procedure of validating the required certificate of two PKIs at the authority level. The chain-of-trust information for the SCMS is described in Section 2.1. We assume a straightforward PKI design for the FL PKI (as there is no proposed PKI for FL)—RCA of FL PKI provides a certificate to ICA of FL PKI by signing the certificate with its own private key while ICA of FL PKI issues a certificate to FLS by signing the certificate with its own private key. We also have FL PKI manager with functionality similar to that of the SCMS manager.

4.2.1 Agreement. During the initial stage, the SCMS manager and FL PKI manager need to come to an agreement about the root of trust of SCMS and FL PKI and share the respective root of trust information with each other and build organizational and technical policy, which ensures efficient and fair operation between SCMS and FL PKI. The root of trust information of SCMS PKI and FL PKI contains all the certificates of electors, RCA of SCMS and RCA of FL PKI respectively. Certificates of electors, RCA of SCMS and FL PKI include the public key of electors, RCA of SCMS and FL PKI respectively. The SCMS Manager shares the certificates of electorsRCA of SCMS with the FL PKI manager, while the FL PKI manager shares the certificate of RCA of FL PKI with the SCMS Manager. The agreement on this root of trust between SCMS PKI and FL PKI information can happen offline, such as physical exchange. It is a potential vulnerability in the initial phase of communication as a lack of proper authentication, encryption, or establishment mechanisms cannot ensure a secure communication system, which must ensure confidentiality, integrity, and authenticity of data. RA and ICA of FL act like a gateway for SCMS and FL PKI respectively during SCMS and FL PKI communication in the authority level. Every message from SCMS and FL PKI is sent by RA and ICA of FL PKI respectively to each other during this authority-level communication. Therefore, the SCMS Manager and FL PKI Manager also share the public key of RA and ICA of FL PKI with each other. Henceforth, the SCMS Manager and FL PKI Manager share the public key of

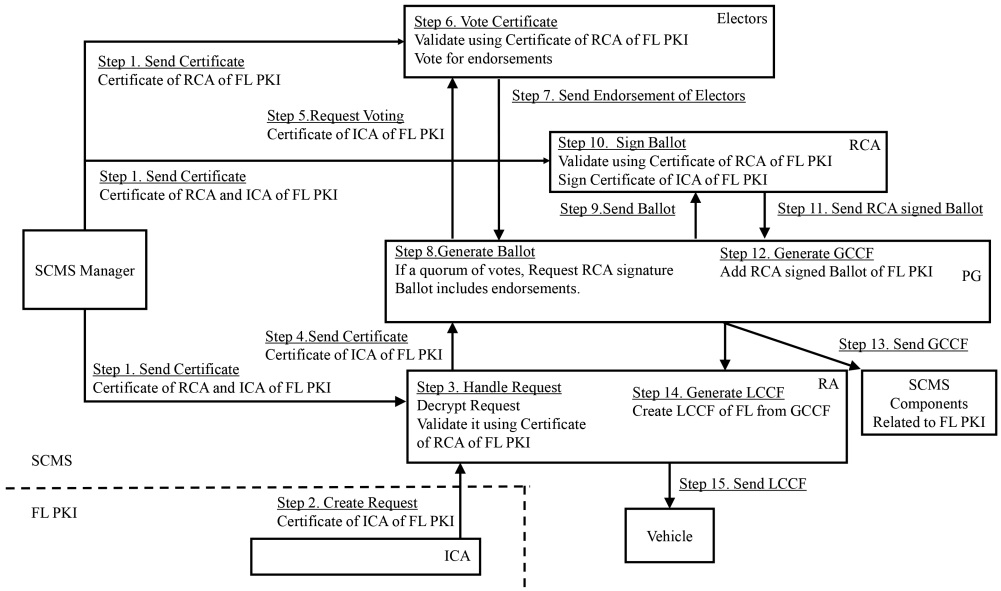


Fig. 3. Certificate provisioning process in SCMS.

ICA of FL PKI and RA to RA and ICA of FL PKI respectively so that RA and ICA of FL PKI can send messages to each other using encryption.

4.2.2 Certificate Provisioning Process in SCMS. In this section, we present a detailed description of the provisioning process of FL PKI certificate chain to vehicles by SCMS. Figure 3 shows this certificate provisioning process in SCMS. The steps are mentioned below:

- **Step 1.** This is the initial setting. SCMS Manager sends a certificate of RCA of FL PKI to electors, RCA of SCMS, and RA. SCMS components require this root of trust information of FL PKI to verify the certificate chain of FL PKI.
- **Step 2.** The certificate of ICA of FL PKI is signed by RCA of FL PKI using its private key and then ICA of FL PKI sends this certificate to RA in SCMS PKI as a certificate validation request by encrypting the request with the public key of RA.
- **Steps 3, 4.** RA decrypts this certificate validation request using the private key of RA and verifies the certificate of ICA of FL PKI. Since RA already knows the public key of RCA of FL PKI, it can verify that the certificate validation request is from the authorized ICA of FL PKI. RA sends the certificate of ICA of FL PKI to PG, encrypting it with the public key of PG and signing it with the private key of RA after successful verification.
- **Step 5.** PG decrypts this message from RA using its private key and checks the authenticity using the public key of RA. Since PG has the communication line to electors, it sends the certificate of ICA of FL PKI to electors, encrypting with a respective public key of electors and signing it with the public key of PG, then PG casts voting for the certificate of ICA of FL PKI to electors.
- **Step 6, 7.** Electors decrypt the request voting message from PG using their respective private keys and validate the authenticity using the public key of PG. Electors know the public key of RCA of FL PKI because of the root of the trust agreement between SCMS PKI and FL PKI. If the certificate of ICA of FL PKI is valid, then electors operate by endorsement as in Table 2. Each elector has its self-generated certificate, and thus the individual public and private key

Table 2. Type of Endorsement

Action	Explanation
Addition of FL PKI	A new chain of FL PKI that is endorsed by a quorum of valid electors can build the trust relationship with other SCMS components.
Revocation of FL PKI	Revoking a chain of FL PKI would stop the operation of all SCMS components related FL PKI.

pair. However, electors send the endorsement (signed certificate of ICA of FL PKI) to PG, encrypting with the public key of PG and signing with their respective private keys.

— **Steps 8, 9.** PG decrypts each endorsement from electors using its private key while validating the authenticity using the public key of electors. If the certificate of ICA of FL PKI gets the least quotum of votes from electors, PG generates a ballot to aggregate all endorsements of electors. Consequently, PG sends it to RCA of SCMS, encrypting it with the public key of RCA of SCMS and signing it with the private key of PG.

— **Steps 10, 11.** RCA of SCMS receives the ballot from PG and decrypts it using its private key while validating the authenticity using the public key of PG. RCA of SCMS can validate the certificate of ICA of FL PKI by the certificate of RCA of FL PKI and validate each endorsement by the signatures of the electors. It is signed by the RCA of SCMS. We refer to this ballot and RCA of SCMS signed FL PKI as *SCMS signed FL PKI*. RCA returns it to PG, encrypting it with the public key of PG and signing it with the private key of RCA of SCMS.

— **Steps 12, 13.** PG decrypts the SCMS signed FL PKI and then assembles updates to the GCCF of *SCMS signed FL PKI* containing all of the SCMS components related to FL PKI, such as electors, RCA, PG, RA, MA, ICA, and PCA. It broadcasts it to SCMS components related to FL PKI.

— **Steps 14, 15.** RA creates LCCF of FL PKI based on the received GCCF of *SCMS signed FL PKI* from PG and then sends it to vehicles. LCCF of FL PKI contains, at a minimum, all of the FL PKI chains that are used to issue FLS for vehicles to support P2P certificate communication.

Note that SCMS shows four types of endorsement (addition of elector or RCA and revocation of elector or RCA). However, we consider adding two types of endorsement as the addition and revocation of the certificate chain of FL PKI. Additionally, an endorsement contains the type of endorsement, the hash ID of the certificate to be endorsed, the generation time of the endorsement, and a signature of the electors [7].

4.2.3 Certificate Provisioning Process in FL PKI. The certificate provisioning process in FL PKI is shown in Figure 4. The FL PKI does not need to have the certificates of all the entities of SCMS. It needs only the certificate of the electors, RCA and ICA of SCMS, and PCA because this certificate chain is required by FLS to verify vehicles. In SCMS, PG sends the GCCF to RA, and RA constructs the LCCF from GCCF and gets it signed by PG. GCCF contains the certificates of all the entities of SCMS, whereas LCCF contains a subset of the certificates in GCCF. In our case, LCCF includes only the certificates of electors, RCA and ICA of SCMS, MA, and PCA. The process is defined as follows:

— **Steps 1, 2, 3.** The FL PKI manager sends a certificate of electors, RCA of SCMS, and RA to RCA and ICA of FL PKI. Those certificates of SCMS are required by these FL PKI components to verify the certificate chain of SCMS. Note that **Steps 2, 3** are the same process as **Steps 13, 14** in the Certificate Provisioning Process in SCMS.

— **Step 4.** RA sends the LCCF to ICA of FL PKI by signing it with its private key and encrypting it with the public key of ICA of FL PKI.

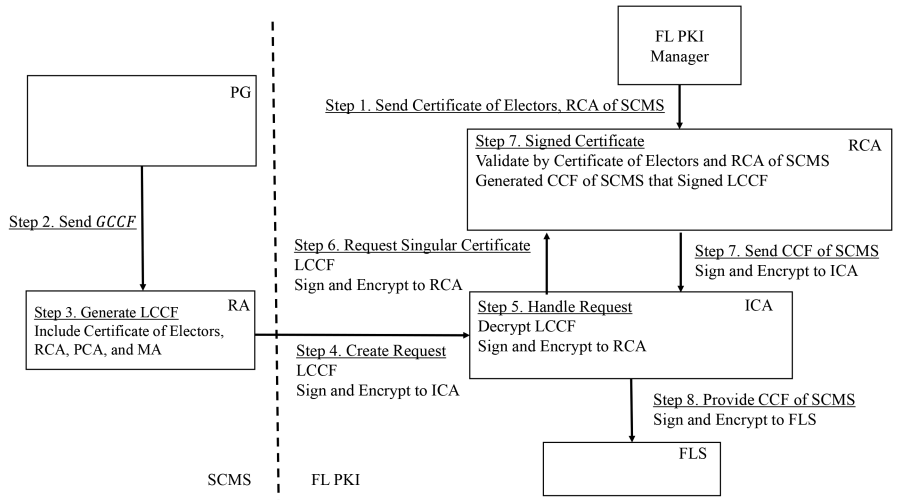


Fig. 4. Certificate provisioning process in FL PKI.

— **Step 5.** ICA of FL PKI decrypts this LCCF from RA using its private key and validates the authenticity using the public key of RA. ICA of FL PKI sends it to RCA by encrypting it with the public key of RCA of FL PKI and signing it with its own private key.
— **Steps 6, 7, 8.** RCA of FL PKI receives this request of signing the LCCF from ICA of FL PKI and verifies the LCCF by validating the public key of electors and RCA of SCMS from the agreement about the root of trust of SCMS and FL PKI. If the LCCF is valid, then RCA of FL PKI signs LCCF with its private key. We refer to this signed LCCF as the **Certificate Chain File (CCF)** of SCMS. It sends it back to the ICA of FL PKI, which in turn sends it to FLS. The CCF of SCMS supports p2p certificate communication between FLS and vehicle.

4.3 Aggregation Protocol Description

Once the agreement and certificate provisioning processes for both SCMS and FL PKI are complete (as described in Section 4.2), both the vehicles and FLS can engage in the secure aggregation protocol. We first define some notations and symbols in Table 1 to describe our protocol. Our federated learning aggregation protocol for vehicular network is shown in Figure 5. In particular, we use $\bar{w}_\tau$ to denote the global model and $\bar{g}_\tau^\iota$ to denote the model update reported by the ιth vehicle in the τth round, where $\iota = 1, 2, \ldots, I$ and $\tau = 1, 2, \ldots, T$. We use c to denote the certificate of FLS and pc_κ^ι to denote the κth pseudonym certificate reported by the ιth vehicle, where $\kappa = 1, 2, \ldots, K$. Note that Vehicle$_\iota$ can use only one pseudonym certificate during a particular round, τ, and multiple pseudonym certificates during the whole secure aggregation procedure. We use MD to denote Misbehavior Detection for FL and AGG to denote the aggregation algorithm that is described in detail in Sections 5 and 2.4, respectively. In this section, we describe our secure aggregation protocol as follows:

— **Step 1.** FLS sends a broadcast message with its certificate, c, at the beginning of each round, τ.
— **Step 2.** Each vehicle, Vehicle$_\iota$ (participating in the secure aggregation protocol), generates a response message to FLS given the condition that c is valid by the LCCF of FL PKI in Vehicle$_\iota$. c is the end-entity of FL PKI. Vehicle$_\iota$ sends a response message to FLS with its pseudonym certificate, pc_κ^ι. FLS registers Vehicle$_\iota$ with pc_κ^ι if it verifies pc_κ^ι through CCF of SCMS.

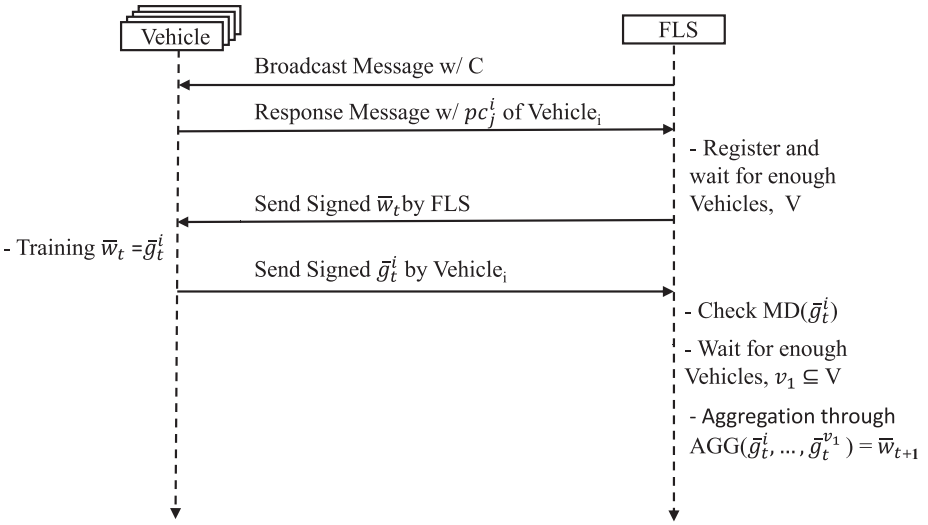


Fig. 5. Our federated learning protocol for vehicular network.

— **Step 3.** FLS waits to send the global model, $\bar{w}_\tau$, of round, τ, until it receives response messages from enough vehicles with the total number of participating vehicles, V, where V is decided by FLS. FLS sends $\bar{w}_\tau$ to Vehicle$_\iota$ after obtaining V.
— **Step 4.** Vehicle$_\iota$ receives $\bar{w}_\tau$ and then deploys it to train $\bar{w}_\tau$ from the local training dataset. Vehicle$_\iota$ gets a global model update, $\bar{g}_\tau^\iota$, and sends it to FLS by encrypting with the public key of FLS (received from c and signing with the private key of pc_κ^ι).
— **Step 5.** Upon receiving $\bar{g}_\tau^\iota$ from Vehicle$_\iota$ with pc_κ^ι, FLS decrypts the message using its private key and verifies PC$_j$. Misbehavior Detection, MD($\bar{g}_\tau^\iota$), is a process to detect poisoning attacks against FL. If it detects malicious activity for pc_κ^ι, FLS processes revocation in Section 5. Once FLS receives enough of $\bar{g}_\tau^\iota$ from vehicles (given the condition $v_1 \subseteq V$), it computes $\bar{w}_{\tau+1}$ from the aggregation algorithm, AGG($\bar{g}_\tau^\iota, \ldots, \bar{g}_\tau^{v_1}$). Note that V - v_1 is the number of dropped vehicle cases including misbehavior or disconnection. FLS follows the same steps from **Step 1** to **Step 5**. The total number of rounds to be performed is dependent on FLS.

5 REVOCATION

In this section, we explain how SCMS and FL PKI cooperate to remove misbehaving end-entities during FL. The misbehavior detection program in FL is different from misbehavior detection in SCMS. There can be different types of misbehavior during federated learning, i.e., attacks related to federated learning such as data poisoning [17] and model poisoning [3]. We suggest the detection methods for misbehavior for FL from the following research papers:

— **FedDetector** [35] detects malicious clients from model update consistency using historical information based on extracted features from model updates in one or multiple rounds. In particular, the server predicts a client's model updates through the client's model consistency for several rounds.
— **Fang Detector** [17] was proposed to detect malicious clients by computing the accuracy of the best global model for FL training rounds. In particular, an attack has effects on reducing the accuracy of the global model.

— **Sybil detector** [16, 19, 28, 33, 34] is used to detect malicious clients. Sybil detection methods use the clients' IPs, misbehavior, and social graphs. In particular, SCMS also provides a blacklist in which MA adds the misbehavior into CRL to contain revoking misbehavior information [7].

Since the detection methods can detect misbehavior or malicious activities for FL, their research [11, 24] has suggested the recovery of poisoning global model update parameters. If it can recover the analogous parameters before poisoning attacks, we will incur less vehicle computation and communication costs. Note that the detection methods of this misbehavior for FL and the model recovery methods for FL are independent of our approach [12].

FLS can detect malicious activity as FL proceeds with vehicles. Thus, there is a need to determine misbehavior and identify those misbehaving Vehicle_ι. Since Vehicle_ι uses pc_κ^ι during FL, FLS cannot verify the real identity of the Vehicle_ι. However, it can still revoke a vehicle by providing pc_κ^ι of the Vehicle_ι to SCMS. In the following, we show a detailed description of the revocation process between the FLS and SCMS.

— **Step 1.** FLS generates a misbehavior report including the Certificate of FLS C, pc_κ^ι of the Vehicle_ι, and misbehavior activity report on the Vehicle_ι. It encrypts and sends the misbehavior report to MA using the public key of MA and signs it with its own private key through CCF of SCMS. Thus, the misbehavior report can only be decrypted by MA. FLS encrypts this misbehavior report again with the public key of RA and sends it to RA.
— **Step 2.** RA decrypts the request using the private key of RA and verifies it with LCCF of FL PKI. However, it cannot read the misbehavior report, which is encrypted using the public key of MA issued by FLS. RA sends the encrypted report to MA.
— **Step 3.** MA decrypts the encrypted report using its private key and processes the revocation report for revoking the vehicle according to SCMS policies explained in detail in [7], which are outside the scope of this article. If the revocation report is valid, then MA adds the vehicle to CRL and sends the updated CRL to FLS. FLS stores the updated CRL and can detect the misbehaving vehicle using that CRL.

On the other hand, vehicles can also report the FLS misbehavior to MA in SCMS and then FL PKI Manager takes the necessary step to revoke a FLS. The steps are described below:

— **Step 1.** Vehicle generates a misbehavior report including pc_κ^ι of the Vehicle_ι, the Certificate of FLS C, and the misbehavior activity report of the FLS. It encrypts misbehavior reports to MA in SCMS. The vehicle sends it to MA via RA.
— **Step 2.** The MA receives misbehavior reports and then runs misbehavior detection algorithms to determine which reported certificate of the FLS may be validated. It encrypts misbehavior reports using the public key of ICA of FL PKI and then sends them to ICA via RA.
— **Step 3.** ICA of FL PKI receives misbehavior reports from RA and then runs misbehavior detection algorithms for FL. It can process revocation of FLS according to FL PKI policies under FL PKI manager. The ICA of FL PKI sends the FLS information to MA, which stores it inside the updated CRL and then automatically updates the CRL for every entity.

6 SECURITY ANALYSES

In this section, we provide the security analysis of our approach. Even though we do not provide formal proof of security, the underlying goal is the confidentiality of the vehicle, the confidentiality of FLS's response, the integrity of the pseudonym certificate, and the unlinkability of the pseudonym certificate. Our proposed approach utilizes the pseudonym certificate scheme from the SCMS for the authentication process between vehicles and FLS.

6.1 Hiding Privacy-sensitive Information from FLS

In our proposed approach, the goal of encrypting the response message with a public key of a pseudonym certificate is to prevent the FLS from learning sensitive information. The pseudonym certificate does not contain any privacy-sensitive information like the End-entity certificate (Vehicle) issued by **Enrollment CA (ECA)** containing privacy-sensitive information.

Theorem. *Security of pseudonym certificate scheme from FLS: Suppose that the FLS follows an honest but curious setting, sending the message using key pairs of pseudonym certificates to FLS. In this case, FLS is unable to verify the actual identity but can verify the pseudonym certificate of the vehicle.*

Proof. We can prove this case straightforwardly. If the encryption is performed with the PKI, these are based on the authenticated encryption to combine confidentiality and integrity guarantees for communication between parties. SCMS and FL PKI build on a trust relationship. One certificate without privacy-sensitive information is issued by SCMS. Another certificate is issued by FL PKI. They can verify each other. A pseudonym certificate in SCMS is unlinkable between FLS and vehicle. □

6.2 Honest but Misbehavior or Malicious Activity for FL

During communication between the vehicle and FLS, we consider an honest activity although it may constitute misbehavior or malicious activity for FL. The vehicle may or may not be compromised and is allowed to provide malicious global model updates or misbehave to manipulate or poison from poisoning attacks.

Theorem. *Security of adversarial attacks for FL: Suppose that the vehicle follows the honest but misbehavior or malicious activity setting and sends the poisoned global model updates to FLS. In that case, it relies on the defense methods of adversarial attacks for FL in terms of Byzantine-robust aggregation algorithms, detection, and recovery methods. It provides privacy of vehicles that face active adversaries.*

Proof. The proof in this case is straightforward because it is based on recent research methods [16, 17, 19, 28, 33–35]. However, to provide more transparency, we detail the assumptions made in this scenario. We assume that FLS deploys a detection program based on methodologies discussed in [16, 17, 19, 35] to identify misbehavior or malicious activities in FL. Additionally, a recovery method [12] is employed to recover the global model from adversarial attacks. The specific details of these assumptions and methods are further elaborated in Section 5. □

6.3 Confidentiality of Misbehavior Report

In the SCMS, the unified butterfly key expansion scheme [25] can prevent the RA from message contents. Encrypted messages using key pairs between vehicle and MA, MA and ICA FL PKI, or MA and FLS remain unknown by the RA because it does not know the private key. Encrypting the MA report message using the private key and signature provides only observation-related protection from entities such as MA and ICA of FL PKI.

Theorem. *Security of misbehavior report by RA: Suppose that the RA follows an honest but curious setting, sending the misbehavior report to MA via RA. In this case, the RA cannot decrypt the misbehavior report using MA and vehicle or FLS key pairs because the private key protects it from recovery. It also signs the misbehavior report with the signature.*

Proof. We start by noticing that encrypting the misbehavior report is used for unified butterfly key expansion including authenticated encryption and a signature scheme [25]. Authenticated

encryption allows encrypting and decrypting messages using key pairs so that it can provide a guarantee for any adversary that encrypts the message. The signature scheme consists of three steps: generating the key pair using the security parameter, signing the message by computing the signature from the private key and the message, and verifying the signature's correctness using the public key, the message, and the signature. In both schemes, RA cannot decrypt it and validate it if modifying the message. □

6.4 Independent Trust Domains

The authorities in each PKI share their information during the certificate chain generation process. If the component in the domain is compromised, it cannot process the policy in another domain because different components and trust domains represent different logical functions. The different trust domains share minimal authorization information through provisioning GCCF of FL PKI and CCF of SCMS. If the component in SCMS or FL PKI is compromised, it is not effective in another trust domain because each domain processes independently.

Theorem. *Security of trust domains: Suppose that there is a domain for a trust authority that is compromised. In this case, the two different trust domains share minimal authorization information through provisioning GCCF of FL PKI and CCF of SCMS. If the component in SCMS or FL PKI is compromised, it is not effective in another trusted domain because each domain processes independently.*

Proof. We consider sharing minimal information about the authorities and policy between two different PKIs. The information about the authorities contains certificates of each authority, including the public key of the respective authorities, and the policy contains service permission for each component. If the component in FL PKI is compromised, it cannot have permission to control the service for other components in SCMS. □

6.5 Privacy for the Eavesdropper

An eavesdropper observing the communication between the vehicle and FLS faces significant challenges in tracking the vehicle due to the utilization of pseudonym certificates. Each vehicle employs a unique pseudonym certificate for every communication with FLS, resulting in frequent changes to the certificate associated with a specific vehicle. Consequently, the pseudonym certificate itself contains no information that can be linked to the vehicle, making it extremely difficult for an eavesdropper to track the vehicle's movements. Note that SCMS provides strong multiple-layered key management and detection programs for attacks, such as collusion attacks and man-in-the-middle attacks.

6.6 Multiple Pseudonym Certificates from the Same Vehicle for Multiple Rounds

In SCMS, one vehicle can have up to 20 pseudonym certificates. A malicious vehicle can send multiple response messages with its multiple pseudonym certificates during one round or multiple rounds. However, FLS cannot recognize that the pseudonym certificates are from the same vehicle and thus proceed with the next steps of that round to compute the aggregated value. Therefore, a vehicle can manipulate the aggregated value by sending its multiple gradients with different pseudonym certificates. In this case, we can utilize LOP to prevent the use of multiple pseudonym certificates by one vehicle during a round. Since LOP knows the source address and location of the vehicles, it can detect a vehicle that sends multiple pseudonym certificates during the same round. Therefore, when a vehicle sends multiple response messages with multiple pseudonym certificates, LOP discards the messages. Moreover, LOP can send a misbehavior report to MA to prevent the vehicle from participating in the secure aggregation protocol. Note that SCMS provides that LOP

discards the identifiable information from the message, e.g., IP address, location information, or any other identifiable information of the vehicle only including the pseudonym certificate.

7 EVALUATION

7.1 Computation and Communication for Vehicles

Computation and communication cost: When each vehicle is asked to send a response message with pc_κ^ι and to compute global model update, $\bar{g}_\tau^\iota$, we introduce computation and communication costs to the vehicle. The computation and communication costs do not depend on which round the vehicle performs and computes. Therefore, we can show the total computation and communication cost, which are obtained by (1) performing the response message with pc_κ^ι per vehicle, which takes *O(nT)* time, where T is the total number of rounds, and (2) computing a global model update per vehicle, which takes *O(mT)* time. Overall, each vehicle's computation and communication costs amount to *O(T(n+m))* over all rounds.

Storage cost: The vehicle must store LCCF of FL PKI, pc_κ^ι, and global model, g_τ^ι, which have *O(G)*, *O(n)*, and *O(m)*, respectively. Overall, total storage cost is *O(G+n+m)*.

7.2 Computation and Communication for Server

Computation cost: The server's computation cost can be broken up into three parts: (1) computing the broadcast message with c, which takes *O(nT)* time; (2) generating the global model, $\bar{w}_\tau$, which takes *O(mT)* time; and (3) performing the aggregation algorithm, *AGG*, assuming a global model has m parameters, needing *O(VmT)* times, where V is number of vehicles. Overall, the server's computation cost is *O(T(n+m+Vm))* over the total number of rounds.

Communication cost: The server's communication cost can be broken up as (1) sending the broadcast message with C, which takes *O(n)* time, and (2) sending the global model, $\bar{w}_\tau$, which takes *O(m)* time. Overall, the server's communication cost is *O(n+m)* and *O(T(n+m))* over the total number of rounds.

Storage cost: The server must store CCF of SCMS, c, which requires *O(G)* and *O(n)* space, respectively, and global model updates, $\bar{g}_\tau^\iota$, which requires *O(mV)* space, where V is the total number of vehicles. Overall, total storage cost is *O(G+n+mV)*.

7.3 Experimental Setup

7.3.1 Network Setting. We perform an FL experiment while driving a vehicle on different types of roads. Figure 6 shows the yellow point as a base station; the local road has a speed limit of 35 miles per hour with signals (point A to point B); the urban road is a highway with a speed limit under 65 miles per hour (point C to point D); and the rural road is a highway with a speed limit over 75 miles per hour (point D to point E). The FL protocol is deployed between a vehicle located in central Colorado and five domestic and five international Google Cloud servers: Los Angeles, South Carolina, Iowa, Oregon, Virginia, Sao Paulo, Sydney, Seoul, Singapore, and London, respectively.

7.3.2 Dataset. We consider multiple datasets for FL processes in our evaluation. We use two image classification datasets, Fashion-MNIST and CIFAR10.

- **Fashion-MNIST [10]** is a 10-class digit image classification dataset that contains 70,000 fashion images split into 60,000 training images and 10,000 test images, where the size of each image is 28 x 28.
- **CIFAR10 [21]** is also a 10-class 32 x 32 RGB image classification dataset. Each class has the same number of samples with 6,000 images per class.

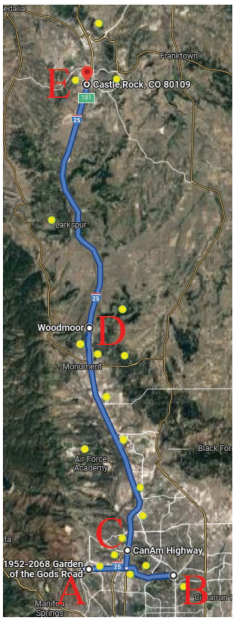


Fig. 6. Driving route and base station.

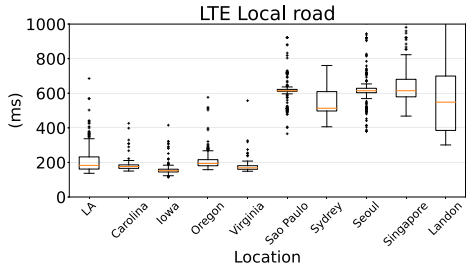
7.3.3 FL Setting. We assume that the vehicles use an SGD optimizer to compute model updates. Considering the different characteristics of the datasets, we set parameters for the original FL training. We train CIFAR10 with Alexnet using a batch size of 250 for 1,200 rounds. We train Fashion-MNIST with Adam optimizer using a batch size of 250 for 1,200 rounds. We consider four aggregation algorithms: Bulyan, Median, Krum, and Trimmed-mean.

7.3.4 Implementation Setting for Vehicles and Server. We describe the implementation setting for FL between vehicles and a server in our approach. We run experiments on 10 cloud-based Linux servers (5 domestic and 5 international servers) in different geographical locations with 3.2 GHz of CPU and 16 GB of RAM. We also run the implementation inside a vehicle using a laptop with 2.80 GHz of CPU and 32 GB of RAM. We consider three types of road conditions (local, urban, and rural) by two types of cellular network environments (LTE and 5G).

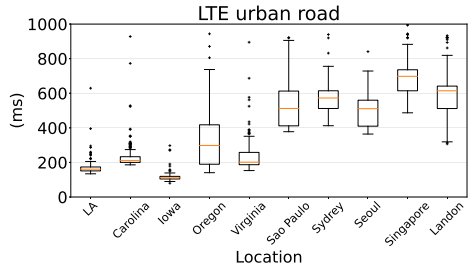
7.3.5 Attack Setting. By default, we assume that 100 vehicles' local training data are non-iid. We sample 10% and 20% of vehicles as malicious vehicles. We choose three types of adversarial attacks for FL, namely Min-Max attack, Min-Sum attack, and LIE attack. Note that all of our attacks outperform the existing attacks. In the Min-Max attack, the malicious vehicle modifies the global model updates to compute the maximum distance from other global model updates. In the Min-Sum attack, a malicious vehicle computes the malicious global model updates to sum squared distance from other benign global model updates. LIE attack is a common adversarial attack on machine learning or FL that adds a small amount of noise to global model updates. This small noise has a huge impact on the aggregation process in FL.

7.4 Experimental Results

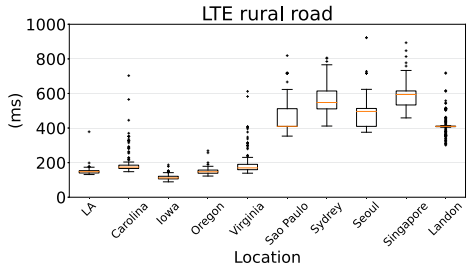
7.4.1 Communication Results. Our work is the first to comprehensively consider communication latency during driving while designing a privacy-preserving protocol for FL in a vehicular network. In this section, we compare the performance while driving on different types of roads.



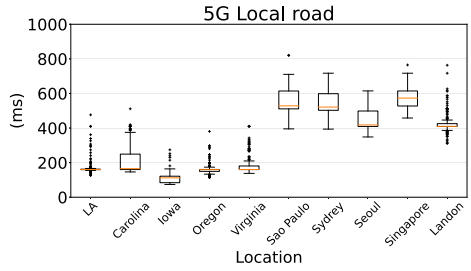
(a) Communication Latency of local roads in LTE between vehicle and 10 servers



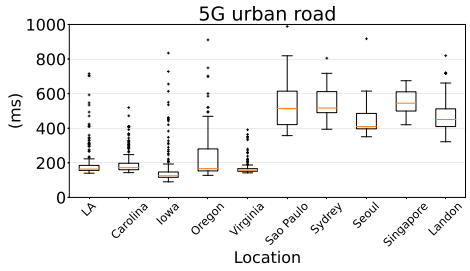
(b) Communication Latency of urban roads in LTE between vehicle and 10 servers



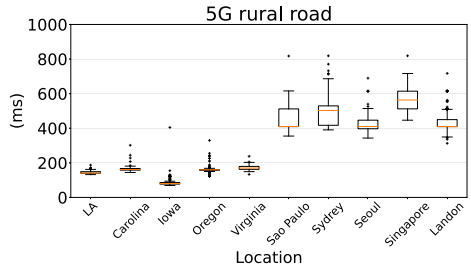
(c) Communication Latency of rural roads in LTE between vehicle and 10 servers



(d) Communication Latency of local roads in 5G between vehicle and 10 servers



(e) Communication Latency of urban roads in 5G between vehicle and 10 servers



(f) Communication Latency of rural roads in 5G between vehicle and 10 servers

Fig. 7. Communication latency between vehicle and 10 geographically separated data centers in the world, executed over cellular network environments (LTE and 5G) on three types of roads during driving.

We collected latency of communication and running time of registration and validation while operating a vehicle on three types of roads. Figure 7 shows the communication results between the vehicle and 10 geographically separated data centers in the world while driving on three types of roads assuming local, urban, and rural environments. The orange line is the median of the total communication latency; the **interquartile range (IQR)** is the box showing the dispersion calculated by subtracting the first quartile in the lower 25% from the third quartile in the upper 75%. The whiskers represent outside of the middle 50%, and out of range as the crossed outlier point. We measure the median as the average value of total communication latency; the IQR indicates the stability of connectivity, the whiskers as less stable connectivity, the crossed outlier point as unstable connectivity, and a combination of the median and IQR showing the skewness of latency.

In Figures 7(a) and 7(d), we analyze the communication latency on the local road, focusing on when there are changes in the geographical locations of servers while using LTE and 5G. The median RTT shows that 5G is approximately 14.5% faster than LTE (domestic servers approximately 13.7% and international approximately 15.4%). The fastest server is Iowa and the slowest server is Oregon, where 5G is approximately 26.4% and 18% faster respectively than LTE among domestic servers. The fastest server is London and the slowest server is Singapore where 5G is approximately 23.4% and 6.8% faster respectively than LTE among international servers. 5G is approximately 33.2% more stable than LTE (domestic servers approximately -8.2% and international approximately 38.4%). Most values of latency show a positive skew.

As seen in Figures 7(b) and 7(e), 5G is approximately 14.8% faster than LTE, with domestic servers approximately 14% and international servers approximately 15.6% faster. Specifically, the fastest server is Iowa and the slowest server is Oregon, where 5G is approximately -12% and approximately 44.2% faster respectively than LTE among domestic servers. The fastest server is Seoul and the slowest server is Singapore, where 5G is approximately 19.8% and approximately 22% faster respectively than LTE among international servers. 5G is approximately 25.8% more stable than LTE with domestic servers approximately 46.8% and international approximately 12.3% faster. Most values of latency have a positive skew. However, international servers on LTE have a negative skew.

Comparing the communication latency of the rural road on LTE (Figure 7(c)) and 5G (Figure 7(f)), we note that 5G is approximately 6.3% faster than LTE (domestic servers approximately 6.3% and international servers approximately 6.2%). The fastest server is Iowa such that 5G is approximately 28.4% faster, and the slowest server is Virginia, which is approximately 1% slower among domestic servers. The fastest server is London and the slowest server is Singapore, where 5G is approximately 1% and approximately 5.3% faster respectively than LTE. 5G is approximately 7.1% more stable than LTE (domestic server approximately 49.5% and international server approximately 3.3%). They have positive skew and normal distributions, but some international servers show negative skew.

Comparing local, urban, and rural roadways, we show that 5G is 11.9% faster than LTE, with domestic servers approximately 11.3% faster and international servers approximately 12.4% faster in total latency. Combined sequentially, rural, local, and urban are faster on 5G than LTE. This is expected, as the frequency of changing the base station, distance between the vehicle and base station, and capacity of connectivity on the base station have more impacts on the communication latency because we change only the conditions of the driving environment and keep the same network setting. To compare urban and rural roads, the rural road is 17.6% and 6.7% faster than the urban road on LTE and 5G respectively. Figures 7(a), 7(d), 7(b), and 7(e) show that the local road is 2.9% and 1% faster than urban on LTE and 5G respectively. It seems similar to network performance. For local and rural roads, the rural road is 15.2% and 6.3% faster than the local road. In Figure 6, the vehicle connects to two to three base stations on a rural road, and the frequency of changing base stations on the urban road is more than on other roads.

Whether domestic or international servers, in our simulation with a given network setting, the number of hops is consistently 10. Figure 9 shows the geographical distance between the vehicle and servers. This is expected, as the geographical distance is relevant to communication latency. As a result, Iowa, Los Angeles, Oregon, South Carolina, and Virginia are faster among domestic servers respectively. London, Sao Paulo, Seoul, Sydney, and Singapore are faster among international servers respectively. Domestic servers are 2.77x faster for LTE and 3.18x faster for 5G than international servers. The domestic servers are almost 3.27x faster for the local road, 3.03x faster for the urban road, and 2.63x for the rural road than international servers, while it is almost 3.32x faster for local roads on LTE and 3.23x faster for local roads on 5G compared to international

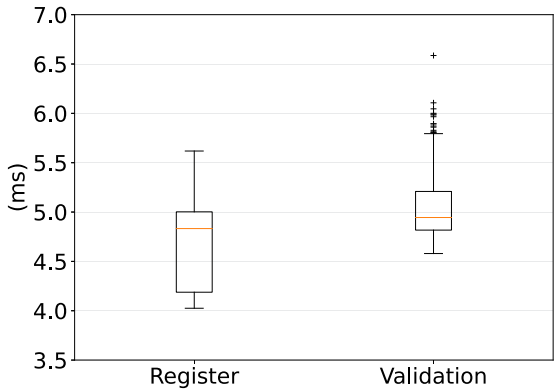


Fig. 8. Running time for registration and validation processes on the server and ignoring communication latency.

servers. Domestic servers are almost 2.96x faster for the urban road on LTE and 3.1x faster for the urban road on 5G, and almost 2.04x faster for the rural road on LTE and 3.22x faster for the rural road on 5G. The crossed outlier point shows that unstable connectivity (e.g., packet loss, connection failures) is like a loss rate for connectivity; the dropout case for our protocol is 9.77% on cellular networks. During the registration process in our FL protocol, the average of dropout rates is added to the total number of participating vehicles, denoted as V. This dropout selection is performed as FL selects a final subset, denoted as v_1, from the total number of participating vehicles.

7.4.2 Federated Learning Results. In Figure 8, we show the running time for the registration and validation processes on the server. The median time of the registration process is 4.83 ms and for the validation process is 4.94 ms on the server. Table 3 shows the running time for training per vehicle using CIFAR10 with Alexnet and for each aggregation algorithm ignoring communication latency. We observe that the vehicles have a somewhat shorter training time than a server because vehicles try to train a small amount of dataset. The global model update size per vehicle is smaller than the aggregation model for the server. The aggregation model is the sum of all vehicles' global model update sizes. In the case of the server, we show a different running time for aggregation algorithms because they use different mathematical formulas. Bulyan and Krum require similar running times to aggregate the global model updates. Increasing the number of vehicles has an impact on the running time of the aggregation protocol. When we increased to 100 from 50 vehicles, the running time for the aggregation algorithm increased by almost 7.30x for Bulyan, 7.73x for Krum, 13.79x for Median, and 4.52x for Trimmed-mean. The aggregation model size is relevant to the number of vehicles.

Figure 10 shows the effect accuracy of increasing the percentage of malicious vehicles using several attacks on Byzantine-robust aggregation algorithms in FL when the round number is varied from 0 to 1,200 for CIFAR10 with Alexnet. For all of the Byzantine-robust aggregation algorithms, the accuracy always reduces with an increasing percentage of malicious vehicles. For instance, 20% of malicious vehicles have an impact of more than 39% on the accuracy compared to 10% of malicious vehicles with Krum in min-max attacks. In the LIE attacks with Krum, 20% of malicious vehicles have an impact of more than 21% on accuracy compared to 10% of malicious vehicles. The result of Byzantine-robust aggregation algorithms in accordance with different attacks is different. Trimmed-mean with LIE attack does not have effects on accuracy, but other attacks reduce the accuracy.

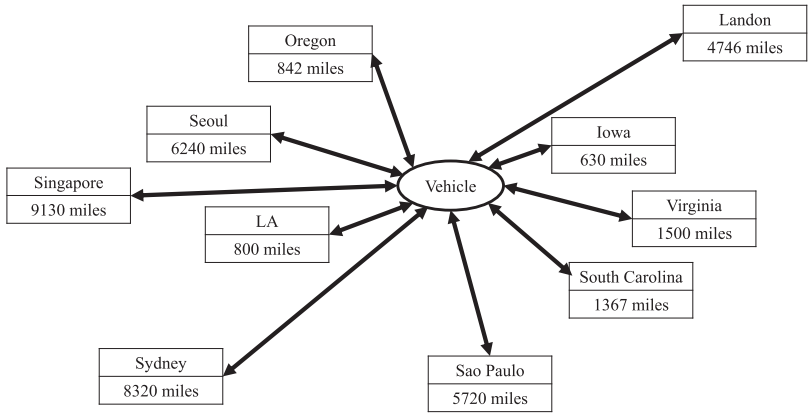


Fig. 9. The distance between vehicle and 10 geographically separated data centers in the world.

Table 3. Running Time for Training per Vehicle for Several Aggregation Algorithms on the Server

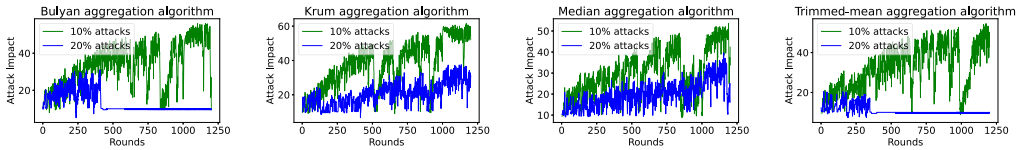
Num. Vehicles	Vehicle				
	Bulyan	Krum	Median	Trimmed-mean	Model Size
1	1.8511 ms	1.9769 ms	2.1383 ms	2.2493 ms	9.42 MB (Fashion-MNIST) 3.88 MB (CIFIR10)
Num. Vehicles	Server				
	Bulyan	Krum	Median	Trimmed-mean	Aggregation size
50	5,281.4630 ms	4,934.3343 ms	0.02641 ms	0.1006 ms	471 MB (Fashion-MNIST) 194 MB (CIFIR10)
100	3,8539.53199 ms	38,139.73197 ms	0.3641 ms	0.4547 ms	942 MB (Fashion-MNIST) 388 MB (CIFIR10)

All running times are for a single-threaded vehicle and server running in 10 geographically separated data centers in the world and only include computation time for training local model per vehicle and aggregation time for each aggregation algorithm on the server. Each entry represents the average over 1200 rounds, with rounds more than 3 standard deviations from the mean discarded.

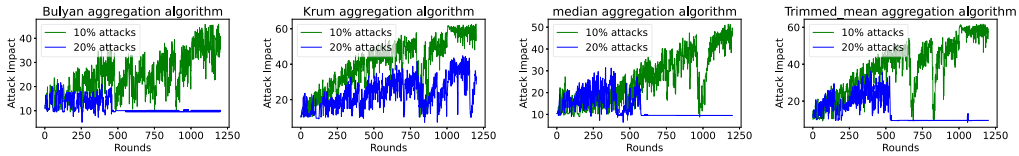
8 DISCUSSION AND LIMITATION

Data authentication and privacy of vehicular and of raw data are essential for effective and acceptable FL for vehicular networks. The challenges include the scalability issue for the pseudonym certificate in SCMS and the stability of connection on the cellular network during driving. In our current study, we focused on driving a single vehicle on three different road types, excluding scenarios involving traffic jams or rush hours. Implementing a real-time federated learning process for multiple vehicles posed a challenge in the current context. If we can collect datasets from vehicles in real time, if datasets from vehicles can be collected in real time, the metrics used to evaluate the performance of FL between the vehicle and server would be more accurate. If we collect various hourly driving experiences in different geographical regions, e,g., mountains using cellular networks, we can improve our work further.

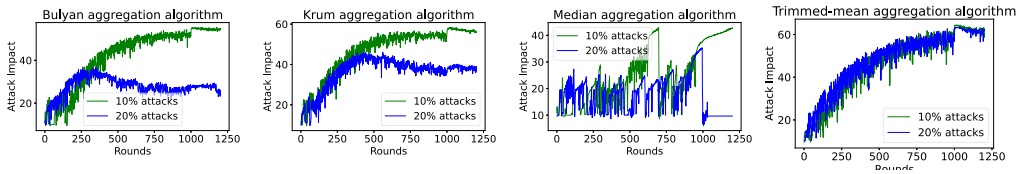
Our approach incurs extra storage and computation costs for the server. Section 7.2 shows that global model updates, $\bar{g}_\tau^t$, require $O(mV)$ time, where V is the total number of vehicles. The server needs $O(mV)$ extra storage to save the global model updates. For instance, if 100,000 vehicles provide the global model updates from mass-driving raw data to the server for 1,000 rounds, the server needs approximately 1,000 TB of extra storage. However, we cannot expect global model updates from mass-driving raw data because Table 3 shows the differences in aggregation sizes depending



(a) Min-max attack on aggregation algorithms



(b) Min-Sum attack on aggregation algorithms



(c) LIE attack on aggregation algorithms

Fig. 10. Effect of increasing the percentage of malicious vehicles using poisoning attacks (min-max attack, min-sum attack, and LIE attack) on the impacts of the accuracy of the existing Byzantine-robust aggregation FL algorithms such as Bulyan, Krum, Median, and Trimmed-mean respectively. The green line is 10% of malicious vehicles and the blue line 20% of malicious vehicles.

on the dataset with the machine learning model. The A2D2 dataset [18] is an autonomous driving open dataset. Its size is roughly 2 TB. The server needs up to 1,000 TB of extra storage. Moreover, the server needs to compute approximately $O(mV)T$ global model updates, where T is the total number of rounds; 1 TB of aggregation model for 1,000 rounds requires 1,000 TB computation cost. Therefore, storage and computation costs are required for the server with power memories.

9 CONCLUSION AND FUTURE WORK

In this article, we introduce a privacy-preserving protocol for FL situated between SCMS and FL PKI to maintain the trust at the authority level, thus leading to trust in end entities of these two orthogonal authorities and the FL process over networks. We utilize SCMS to maintain vehicular privacy during the communication of vehicles with an FLS and FL to maintain the privacy of raw data to FLS. We use 5G and LTE to measure the distance between vehicles and FLS geographically, latency to compute the latency requirement with respect to the distance, and revocation from misbehavior detection in our approach. An interesting future work would be to extend misbehavior detection methods for FL or recovery methods from poisoning. Specifically, misbehavior detection using Explainable Artificial Intelligence among the historical learning dataset will show processes inside machine learning that can be designed and checked to detect poisoning attacks in malicious clients. Another interesting future work is an exploration of recovery methods using machine learning while increasing the percentage of malicious clients on the Byzantine-robust aggregation algorithm. We require only FL in a vehicular network. Our scheme between SCMS and FL PKI has

immediate orthogonal authority to other vehicular applications, i,e., the Internet of Vehicle, and we expect to apply it in other domains in the future.

ACKNOWLEDGEMENT

This work builds on our previous conference paper [9] at the Silicon Valley Cybersecurity Conference 2023. We have significantly improved and extended the conference paper for this journal manuscript. These extensions and additions include a more comprehensive and rigorous analysis of the federated learning process (e.g., analyzing adversarial attacks for federated learning), the implementation and experimental analyses of federated learning in a cloud environment to simulate more realistic networking environments, and the applications of our scheme of federated learning to a vehicular network.

REFERENCES

[1] 3GPP. TS 23.287. 2021. Architecture Enhancements for 5G System (5GS) to Support Vehicle-to-Everything (V2X) Services.

[2] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2020. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2938–2948.

[3] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. 2019. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*. PMLR, 634–643.

[4] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. 2017. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in Neural Information Processing Systems* 30 (2017). https://papers.nips.cc/paper_files/paper/2017/hash/f4b9ec30ad9f68f89b29639786cb62ef-Abstract.html

[5] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 1175–1191.

[6] Nader Bouacida and Prasant Mohapatra. 2021. Vulnerabilities in federated learning. *IEEE Access* 9 (2021), 63229–63249.

[7] Benedikt Brecht and Thorsten Hehn. 2019. A security credential management system for V2X communications. In *Connected Vehicles*. Springer, 83–115.

[8] Joakim Brorsson, Paul Stankovski Wagner, and Martin Hell. 2018. Guarding the guards: Accountable authorities in vanets. In *2018 IEEE Vehicular Networking Conference (VNC'18)*. IEEE, 1–4.

[9] SangHyun Byunx, Arijet Sarker, Ken Lew, Jugal Kalita, and Sang-Yoon Chang. 2023. Privacy-preserving trust management for vehicular communications and federated learning. In *2023 Silicon Valley Cybersecurity Conference (SVCC'23)*. IEEE, 1–8.

[10] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. 2018. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097* (2018).

[11] Xiaoyu Cao, Jinyuan Jia, Zaixi Zhang, and Neil Zhenqiang Gong. 2022. Fedrecover: Recovering from poisoning attacks in federated learning using historical information. *arXiv preprint arXiv:2210.10936* (2022).

[12] Xiaoyu Cao, Jinyuan Jia, Zaixi Zhang, and Neil Zhenqiang Gong. 2023. Fedrecover: Recovering from poisoning attacks in federated learning using historical information. In *2023 IEEE Symposium on Security and Privacy (SP'23)*. IEEE, 1366–1383.

[13] Hongyan Chang, Virat Shejwalkar, Reza Shokri, and Amir Houmansadr. 2019. Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer. *arXiv preprint arXiv:1912.11279* (2019).

[14] Chang-Wu Chen, Sang-Yoon Chang, Yih-Chun Hu, and Yen-Wen Chen. 2017. Protecting vehicular networks privacy in the presence of a single adversarial authority. In *2017 IEEE Conference on Communications and Network Security (CNS'17)*. IEEE, 1–9.

[15] Tim Dierks and Eric Rescorla. 2008. *The Transport Layer Security (TLS) Protocol Version 1.2*. Technical Report.

[16] John R. Douceur. 2002. The Sybil attack. In *International Workshop on Peer-to-peer Systems*. Springer, 251–260.

[17] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. 2020. Local model poisoning attacks to {Byzantine-Robust} federated learning. In *29th USENIX Security Symposium (USENIX Security'20)*. 1605–1622.

[18] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S. Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, Tiffany Fernandez, Martin Jänicke, Sudesh Mirashi, Chiragkumar Savani, Martin Sturm, Oleksandr Vorobiov, Martin Oelker, Sebastian Garreis, and Peter Schuberth. 2020. A2D2: Audi autonomous driving dataset. CoRR abs/2004.06320, (2020). Retrieved from https://arxiv.org/abs/2004.06320

[19] Neil Zhenqiang Gong, Mario Frank, and Prateek Mittal. 2014. Sybilbelief: A semi-supervised learning approach for structure-based Sybil detection. *IEEE Transactions on Information Forensics and Security* 9, 6 (2014), 976–987.
[20] Songyang Han, Jie Fu, and Fei Miao. 2019. Exploiting beneficial information sharing among autonomous vehicles. In *2019 IEEE 58th Conference on Decision and Control (CDC'19)*. IEEE, 2226–2232.
[21] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009). https://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf
[22] Luis Muñoz-González, Kenneth T. Co, and Emil C. Lupu. 2019. Byzantine-robust federated machine learning through adaptive model averaging. *arXiv preprint arXiv:1909.05125* (2019).
[23] Shiva Raj Pokhrel and Jinho Choi. 2020. Improving TCP performance over WiFi for internet of vehicles: A federated learning approach. *IEEE Transactions on Vehicular Technology* 69, 6 (2020), 6798–6802.
[24] Virat Shejwalkar and Amir Houmansadr. 2021. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *Network and Distributed System Security (NDSS)*.
[25] Marcos A. Simplicio, Eduardo Lopes Cominetti, Harsh Kupwade Patil, Jefferson E. Ricardini, and Marcos Vinicius M. Silva. 2018. The unified butterfly effect: Efficient security credential management system for vehicular communications. In *2018 IEEE Vehicular Networking Conference (VNC'18)*. IEEE, 1–8.
[26] Muhammad Naeem Tahir, Pekka Leviäkangas, and Marcos Katz. 2022. Connected vehicles: V2V and V2I road weather and traffic communication using cellular technologies. *Sensors* 22, 3 (2022), 1142.
[27] Virendra Kumar and Benedikt Brecht. 2018. SCMS PoC Supported V2X Applications. Retrieved February 24, 2022 from https://wiki.campllc.org/display/SCP/SCMS+PoC+Supported+V2X+Applications
[28] Gang Wang, Bolun Wang, Tianyi Wang, Ana Nika, Haitao Zheng, and Ben Y. Zhao. 2018. Ghost riders: Sybil attacks on crowdsourced mobile mapping services. *IEEE/ACM Transactions on Networking* 26, 3 (2018), 1123–1136.
[29] William Whyte, André Weimerskirch, Virendra Kumar, and Thorsten Hehn. 2013. A security credential management system for V2V communications. In *2013 IEEE Vehicular Networking Conference*. IEEE, 1–8.
[30] Wikipeida. 2024. Public Key Infrastrucutre. Retrieved April 7, 2024 from https://en.wikipedia.org/wiki/Public_key_infrastructure
[31] Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. 2018. Generalized byzantine-tolerant SGD. *arXiv preprint arXiv:1802.10116* (2018).
[32] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. 2018. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*. PMLR, 5650–5659.
[33] Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham Flaxman. 2006. Sybilguard: Defending against Sybil attacks via social networks. In *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. 267–278.
[34] Dong Yuan, Yuanli Miao, Neil Zhenqiang Gong, Zheng Yang, Qi Li, Dawn Song, Qian Wang, and Xiao Liang. 2019. Detecting fake accounts in online social networks at the time of registrations. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 1423–1438.
[35] Zaixi Zhang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. 2022. FLDetector: Defending federated learning against model poisoning attacks via detecting malicious clients. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2545–2555.

Received 4 August 2023; revised 11 February 2024; accepted 3 April 2024