"Secure and Differentially Private Hierarchical Federated Learning for Healthcare with Validator Committees" **OR** Byzantine-Resilient Hierarchical Federated Learning with Proof-of-Work and Differential Privacy **OR** "Byzantine-Resilient Hierarchical Federated Learning with Committee Validation and Privacy-Preserving Model Access **OR** "Byzantine-Resilient Hierarchical Federated Learning with Committee Validation"

le.saidi

September 2025

1 Introduction

Machine learning (ML) and deep learning are transforming fields such as healthcare, finance, transportation, and computer vision [**?**]. They enable predictive diagnostics, personalized medicine, fraud detection, and intelligent systems. Their success depends on large, diverse datasets, which often contain sensitive information. This dependence creates major challenges related to privacy, security, and regulatory compliance, especially in healthcare and biomedical research.

Federated learning (FL) addresses these concerns by allowing multiple clients to collaboratively train models without sharing raw data. As shown in Figure **??**, clients compute updates locally and send them for aggregation, which reduces the risk of direct data exposure. This design aligns with privacy laws such as the General Data Protection Regulation (GDPR) [**?**] and the Health Insurance Portability and Accountability Act (HIPAA) [**?**].

However, FL is not free from vulnerabilities. Gradients exchanged during training may leak information through inference attacks such as gradient inversion or membership inference [**?**, **?**]. The aggregation server, often assumed to

be fully trusted, can also be compromised or behave maliciously, leading to tampered updates or model poisoning. In addition, FL is exposed to **Byzantine attacks**, where adversarial participants deliberately send corrupted or poisoned updates to disrupt the training process. Such risks undermine both privacy and model reliability, making robust security mechanisms essential in healthcare contexts.

Several cryptographic and distributed techniques have been introduced to address these issues. Attribute-Based Encryption (ABE) provides fine-grained access control but introduces challenges in scalability and key management [**?**, **?**]. Secret sharing, such as Shamir's scheme, distributes sensitive values across multiple parties, reducing the risk of single-point compromise [**?**]. Multiparty computation (MPC) provides strong privacy guarantees, though at high communication costs [**?**, **?**, **?**]. Each approach improves specific aspects of privacy and trust, but none individually addresses the practical challenges of FL in healthcare.

This work focuses on Federated Learning for Internet of Medical Things (IoMT) environments, where patient data are sensitive and devices are distributed. We propose a hierarchical, privacy-preserving FL framework that integrates Ciphertext-Policy Attribute-Based Encryption (CP-ABE), Shamir's secret sharing, differential privacy, proof-of-work for Sybil resistance, and a rotating validator committee for Byzantine robustness. Fog nodes handle partial aggregation, while a randomly selected leader server performs global aggregation under the supervision of the trusted authority. This integrated design strengthens privacy protection, access control, and Byzantine fault tolerance, making it suitable for secure deployment in healthcare systems.

Motivating Scenario:

Patient data privacy is critical in healthcare due to regulations such as HIPAA and GDPR. Hospitals and research centers aim to collaborate on predictive models for early disease detection and personalized treatment. However, direct data sharing is prohibited to protect sensitive patient information and comply with regulatory frameworks.

A hierarchical federated learning framework enables hospitals to train models locally on their private datasets, sharing only encrypted or privacy-preserving model updates instead of raw patient records. Fog nodes coordinate partial aggregation, and a global coordinator oversees secure distribution and final aggregation of models. This architecture enables scalability while maintaining strict privacy guarantees.

Key threats include inference attacks, where adversaries attempt to reconstruct patient information from model updates, and Sybil attacks, where malicious entities create multiple fake identities to disrupt training. Additional risks involve Byzantine participants providing corrupted updates and communication interception that could leak sensitive metadata.

To address these threats, differential privacy adds noise to model updates, preventing reconstruction of sensitive data. Shamir secret sharing secures aggregation by distributing secrets across multiple validators, making collusion

difficult. A Proof-of-Work mechanism ensures Sybil resistance during registration, allowing only legitimate facilities to join. Rotating validator committees, selected based on reputation and role separation, validate secret shares and authenticate updates through consensus voting. Ciphertext-Policy Attribute-Based Encryption (CP-ABE) enforces fine-grained access control, allowing only authorized institutions to access the aggregated global model.

This integrated cryptographic and trust-aware design ensures patient privacy, regulatory compliance, and resilience against adversarial behaviors. It allows healthcare institutions to collaboratively build accurate models for diagnostics and treatment while preserving data confidentiality and integrity.

Challenges:
Motivated by the above observations, this paper addresses the following key challenges:

- **Data Privacy:** Ensuring the confidentiality of sensitive patient data is critical. Even with hierarchical FL, participants may infer private information from shared updates. The system must integrate robust mechanisms, such as differential privacy and Shamir secret sharing, to prevent data leakage while maintaining learning utility.
- **Model Integrity and Byzantine Fault Tolerance:** Protecting the integrity of local and aggregated models from malicious clients or compromised fog nodes is essential. Ensuring correctness and resisting Byzantine attacks requires validator committees and consensus-based verification of secret shares.
- **Sybil and Spam Resistance:** Malicious participants could create multiple fake identities to manipulate training. Lightweight Proof-of-Work challenges during registration are needed to ensure only legitimate entities participate in the federation.
- **Communication Efficiency:** Distributing model updates among numerous participants and fog nodes can generate substantial overhead. Efficient aggregation strategies and secret sharing schemes are required to minimize communication while preserving privacy.
- **Client and Fog Heterogeneity:** Clients and fog nodes have varying computational capacities, network reliability, and data quality. The framework must handle this heterogeneity to ensure consistent global model convergence.
- **Scalability and Robustness:** As the number of participants grows, maintaining system performance, secure aggregation, and privacy guarantees becomes increasingly challenging. The hierarchical architecture with fog-based partial aggregation and rotating validator committees addresses these concerns.

By tackling these challenges, the proposed framework aims to establish a secure, private, efficient, and scalable hierarchical federated learning system suitable for healthcare and other privacy-sensitive domains.

Contribution:

Our contributions can be summarized as follows:

- **Three-Tier Hierarchical Federated Learning:** We extend hierarchical federated learning by introducing a three-tier structure involving healthcare facilities, fog nodes, and a leader server. This design enhances scalability, fault tolerance, and efficiency in large-scale collaborative learning networks.
- **Enhanced Privacy with Differential Privacy:** We integrate differential privacy into local model updates, adding noise to gradients before aggregation. This provides a formal privacy guarantee, protecting sensitive patient information during model training while allowing accurate learning.
- **Enhanced Security with Shamir Secret Sharing:** We use Shamir's secret sharing to split model updates into multiple fragments distributed across fog nodes. This prevents individual entities from reconstructing the full model, offering an additional layer of privacy and resilience against collusion.
- **Proof-of-Work for Sybil Resistance:** We design a Proof-of-Work mechanism as a lightweight Sybil-resistance layer during the registration process. This prevents malicious entities from creating multiple fake identities and ensures that only legitimate healthcare facilities can participate in training.
- **Rotating Validator Committee:** We introduce a rotating validator committee, selected based on reputation and role separation, to validate secret shares and authenticate updates using digital signatures and consensus voting. This mechanism strengthens Byzantine fault tolerance and maintains trustworthiness of the aggregation process.

2 System Model and Framework

In this section, we provide a clear description of the system architecture and explicitly outline the threat model and security requirements.

2.1 System Model

Figure 1 illustrates the architecture and operational methodology of our hierarchical federated learning system. The system comprises four primary components: a Trusted Authority (TA), distributed fog nodes, validator committees, and healthcare facilities (participants).

- **Trusted Authority (TA):** The Trusted Authority initializes the system by performing cryptographic setup and distributing necessary keys. It generates and issues CP-ABE keys to enforce fine-grained access control and manages participant registration. The TA also coordinates the final global aggregation while ensuring compliance with regulatory standards such as HIPAA and GDPR.

- **Fog Nodes:** Fog nodes act as intermediate aggregators between healthcare facilities and the global coordinator. They perform partial aggregation of local model updates using FedAvg, reducing communication overhead and improving scalability. A leader server, randomly selected among fog nodes, is responsible for global aggregation by combining results from all fog nodes.

- **Validator Committee:** A dynamically selected committee of participants ensures trust management and validation. The committee verifies secret shares transmitted by participants before broadcasting approved shares to fog nodes. This mechanism prevents malicious submissions and enforces Byzantine fault tolerance.

- **Healthcare Facilities (Participants):** Healthcare institutions act as clients holding private medical datasets. Each facility trains a local model on its data and generates secret shares of the updates using Shamir's Secret Sharing. These shares are securely transmitted to the validator committee for validation before being used in aggregation. Differential privacy is applied locally to protect against inference attacks.

The proposed framework ensures secure and scalable federated learning. Initially, the TA registers participants and distributes cryptographic material. Healthcare facilities perform local training and protect their model updates with differential privacy. Shamir's Secret Sharing is used to split sensitive information into shares, which are transmitted to the validator committee. After validation, approved shares are aggregated at fog nodes, and the leader fog node performs the global aggregation. The final encrypted global model is distributed back to participants, with access controlled through CP-ABE policies defined by the TA. This iterative process continues until model convergence.

The integration of these mechanisms ensures confidentiality, integrity, and robustness against malicious participants while maintaining compliance with privacy regulations.

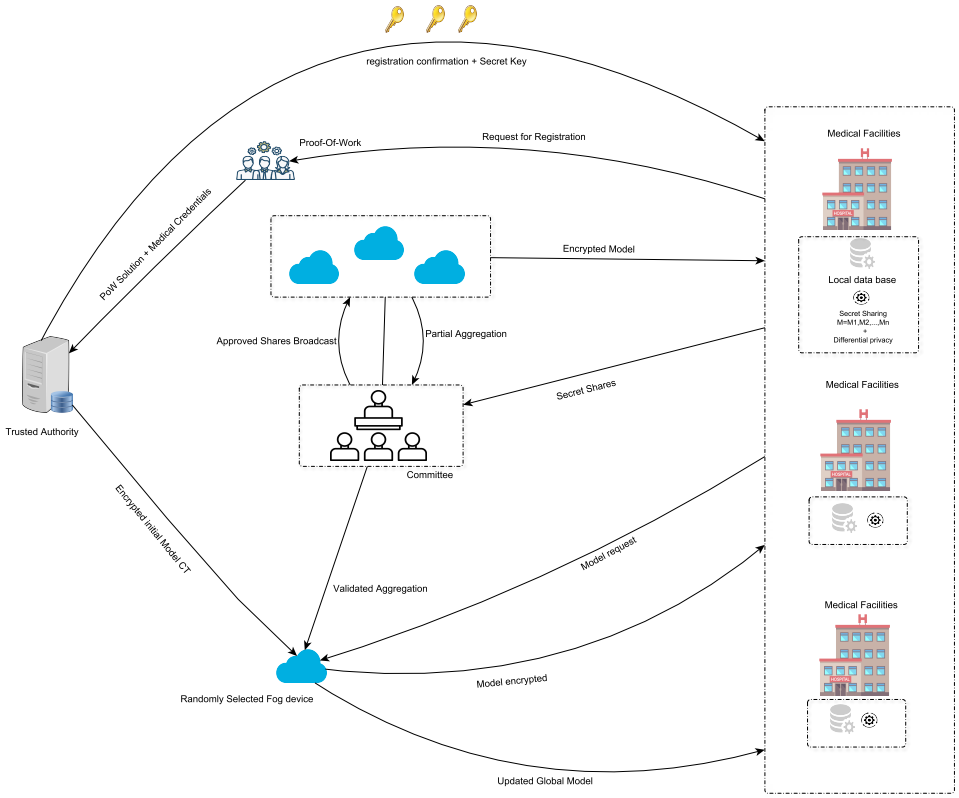

Figure 1: System architecture of the proposed federated learning framework

2.2 Threat Model

The federated learning environment considered in this work is subject to several security and privacy threats. We outline the following adversarial capabilities and attack scenarios:

- **Inference Attacks:** Adversaries may attempt to infer sensitive patient information from shared model updates. Gradient leakage, membership inference, and reconstruction attacks are considered within scope. These attacks can be mounted by both internal participants and external eavesdroppers.
- **Byzantine Attacks:** Malicious participants may submit poisoned or corrupted updates with the goal of disrupting the training process, biasing the global model, or reducing system accuracy. Such adversaries may act individually or collude with others to increase their impact.
- **Sybil Attacks:** A single adversary may create multiple fake identities or compromised clients to gain disproportionate influence in the aggregation

process. This increases the risk of model corruption and reduces fairness in participant selection.

- **Collusion:** Groups of adversarial participants may collaborate to reconstruct sensitive model updates or undermine the integrity of secret sharing. This collusion can amplify inference risks and reduce the trustworthiness of the aggregation process.
- **Communication Threats:** Adversaries may attempt to intercept, modify, or replay communications between participants, fog nodes, and the Trusted Authority. Without secure channels, these actions can compromise confidentiality and availability.

2.3 Security Assumptions and Requirements

In our framework, the following security assumptions are made for the main roles:

- **Trusted Authority (TA):** The TA is assumed to be fully trusted and operates honestly. It is responsible for initializing the system, generating cryptographic keys, enforcing access policies, and distributing the encrypted global model. The security of the system relies on the confidentiality and integrity of the TA's operations.
- **Fog Nodes:** Fog nodes are considered honest but curious. They correctly perform partial aggregation of model updates using FedAvg, but may attempt to learn information from the updates. To mitigate this risk, only secret shares and differentially private updates are handled by fog nodes.
- **Participants (Healthcare Facilities):** Participants are assumed honest-but-curious. They train local models on sensitive patient data and generate secret shares of their updates. They may attempt to extract additional information from the global model, but Shamir's secret sharing and differential privacy ensure that individual data remains confidential.
- **Validator Committee:** Committee members are semi-trusted. They verify secret shares, validate updates, and monitor malicious behavior. They may attempt collusion, but rotating committees and role separation reduce this risk.
- **Leader Server:** The leader server is randomly selected from fog nodes for each global aggregation round. It is assumed to follow the protocol but is still monitored through validation to mitigate potential misbehavior.
- **Network Communication:** Channels are assumed to be vulnerable to eavesdropping or interception. Therefore, all transmitted updates, shares, and model exchanges are encrypted and authenticated.

Security Model Coverage:
The security model addresses the following aspects:

- **Data Confidentiality:** Sensitive patient data remains local and never leaves healthcare facilities. Shamir's secret sharing and differential privacy guarantee the confidentiality of shared updates.
- **Data Integrity:** The validator committee ensures the integrity of updates and secret shares before aggregation, preventing tampering or poisoning attempts.
- **Authentication:** Proof-of-work registration and digital signatures guarantee that only legitimate facilities and fog nodes can participate in training and aggregation.
- **Access Control:** Ciphertext-Policy Attribute-Based Encryption (CP-ABE) enforces fine-grained decryption rights, ensuring only authorized participants can access the global model.
- **Byzantine Robustness:** The system tolerates malicious or corrupted updates through a committee-based validation and monitoring mechanism. The committee, composed of participants selected in a rotating manner, verifies the integrity and consistency of local contributions before aggregation. This approach limits the impact of adversarial behavior, reduces the risk of model poisoning, and strengthens the Byzantine robustness of the proposed framework.
- **Sybil Resistance:** Proof-of-work and reputation mechanisms limit the ability of adversaries to create multiple fake identities.
- **Availability:** Redundancy across fog nodes and committee rotation maintain continuous system operation even under targeted failures or attacks.

These assumptions and requirements collectively ensure that the proposed federated learning framework achieves privacy, integrity, scalability, and resilience in a healthcare environment.

2.4 Framework

The proposed FL framework integrates Shamir's secret sharing, differential privacy, proof-of-work, fog-based aggregation, CP-ABE access control, and validator committees to ensure secure, privacy-preserving, and scalable model training in healthcare environments. Each component addresses challenges related to data confidentiality, communication efficiency, trust management, and robustness against Byzantine and Sybil attacks. The following section provides a detailed description of the framework architecture and its main processes.

2.4.1 System Initialization

In the initialization phase, the Trusted Authority (TA) creates the foundation of the federated learning framework. The TA executes the setup procedure of Ciphertext-Policy Attribute-Based Encryption (CP-ABE), generating a public key (PK) and a master secret key (MSK). The public key is shared with all participants, while the master secret key is kept strictly confidential by the TA. At the same time, the TA defines the set of registered medical facilities U and establishes the attribute universe A, which contains categorical descriptors such as role, institution type, or region. These attributes later control which participants are allowed to access and decrypt the distributed models. The process can be summarized by the following equation:

$$Setup(1^k, U, A) \rightarrow (PK, MSK)$$

where k is the system's security parameter, U represents the set of facilities, and A denotes the attribute universe. Once the setup is complete, the TA generates a private decryption key for each registered facility based on its attributes. This guarantees that only facilities meeting specific attribute-based policies can decrypt encrypted models in the later phases. To prevent malicious entities from flooding the system with fake identities (Sybil attacks) or spamming registration requests, every facility must solve a Proof-of-Work (PoW) challenge before registration. The PoW mechanism requires the facility to compute a valid nonce that, when hashed with its identity and public key, produces a digest below a predefined difficulty target. This ensures that the facility expends real computational resources to prove its legitimacy. The PoW condition is expressed as:

$$H(\text{nonce} \parallel Facility\_ID \parallel PK) \leq Target$$

Here, H represents a secure hash function, nonce is the random number adjusted by the facility, $Facility\_ID$ is its unique identifier, and PK is the system's public key. The Target defines the difficulty level chosen by the TA. Facilities repeatedly adjust the nonce until the condition is satisfied. This procedure works as follows:

- The TA issues a challenge by specifying the difficulty target.
- The facility repeatedly computes the hash of $(nonce||Facility\_ID||PK)$.
- If the resulting hash is greater than the target, the facility changes the nonce and retries.
- Once a valid nonce is found, the facility sends the solution to the TA.
- The TA verifies the hash in constant time, ensuring that the facility actually performed the computation.

Through this mechanism, the system ensures that every registered participant has a computational cost associated with its identity, making Sybil attacks impractical. Only after passing this challenge does the TA provide the facility with its attribute-based secret key. The setup and PoW registration procedures can be illustrated by Algorithm 1 and Algorithm 2, respectively.

Algorithm 1 System Setup by Trusted Authority (TA)

Input: Security parameter λ, set of facilities U, attribute universe A
Output: Public Key PK, Master Secret Key MSK

1: $(PK, MSK) \leftarrow \text{Setup}(\lambda, U, A)$
2: TA securely stores MSK and distributes PK to all registered facilities
3: Define attribute-based access policies for model decryption

Algorithm 2 Facility Registration with Proof-of-Work

Input: Facility request Req, difficulty parameter D
Output: Verified facility F and issued secret keys

1: Facility solves PoW challenge: $Nonce \leftarrow \text{FindNonce}(H(Req||Nonce) < D)$
2: **if** PoW verified by TA **then**
3: TA issues secret key $SK_{facility}$ based on CP-ABE attributes
4: Register facility in system database
5: **end if**

2.4.2 Model Distribution

After initialization, the Trusted Authority (TA) prepares and distributes the initial machine learning model across the federation in a secure and controlled manner. The TA encrypts the global model using Ciphertext-Policy Attribute-Based Encryption (CP-ABE), ensuring that only facilities whose attributes satisfy the access policy can decrypt and use the model. The encryption process enforces fine-grained access control while preventing unauthorized entities from gaining access, even if they intercept the ciphertext. The encryption algorithm is expressed as:

$$Encrypt(Model, PK, T) \rightarrow CT$$

where Model is the initial global model, PK is the public key from the setup phase, and T is the access policy tree that encodes attribute-based restrictions (e.g., $facility_{type}$ = hospital AND region = North). The output CT is the encrypted ciphertext of the model. When a registered facility wishes to obtain the global model, it must first authenticate itself by sending an encrypted request

to the Leader Server (randomly selected from among fog nodes). This ensures secure communication between facilities and fog infrastructure. The encrypted request is computed as:

$$Encrypt(PK_{Leader}, req) \rightarrow encryReq$$

where req is the facility's request for the model and PK_{Leader} is the public key of the leader server. Upon receiving encryReq, the leader server decrypts the request to recover req. Once verified, the leader server transmits the CP-ABE ciphertext CT to the requesting facility. Only facilities whose attributes satisfy the policy T can decrypt the ciphertext. The decryption process is:

$$Decrypt(CT, SK_{user})) \rightarrow Model$$

where SK_{user} is the private decryption key generated by the TA for that facility during initialization. This process ensures several guarantees:

- Confidentiality – Only facilities with valid attributes can decrypt and access the model.
- Controlled Distribution – Unauthorized entities, even if they intercept the ciphertext, cannot recover the model.
- Scalability – Multiple facilities can request the model simultaneously from the fog-based leader server without overloading the TA.
- Authentication – Facilities prove their legitimacy by encrypting their requests to the leader server, which prevents replay or impersonation attacks.

As a result, the model distribution stage establishes a secure channel between the federation coordinator and the medical facilities, ensuring that only legitimate and policy-compliant participants begin local training.

The secure distribution of the initial global model, including CP-ABE encryption and request authentication, is detailed in Algorithm 3.

Algorithm 3 Initial Model Distribution and Decryption by Facility

Input: Encrypted initial model CT, facility private key $SK_{facility}$
Output: Decrypted initial model M_{init}

1: $M_{init} \leftarrow$ User_Decrypt$(CT, SK_{facility})$
2: Facility uses M_{init} as the starting point for local training

2.4.3 Local Training, Differential Privacy, and Secret Sharing

After receiving the encrypted global model from the leader server, each medical facility decrypts it using its private key obtained during system initialization.

The decryption is performed as:

$$User\_Decrypt(CT, SK_{facility}) \rightarrow Model$$

where CT is the encrypted global model and $SK_{facility}$ is the facility's private secret key. If the facility's attributes satisfy the CP-ABE policy, the decryption succeeds. Once the global model is obtained, each facility performs local training on its private dataset D_i for a specified number of epochs E. The result of this computation is an updated local model M_{local_i}. To ensure privacy, before sharing, each facility perturbs its local model with differential privacy noise. Using the Gaussian mechanism, the facility computes:

$$M'_{local_i} = M_{local_i} + \mathcal{N}(0, \sigma^2)$$

where $N(0, \sigma^2)$ is Gaussian noise with variance σ^2, calibrated to the privacy budget (ϵ, δ). This ensures that sensitive patient-level information cannot be inferred from the model updates. Next, the differentially private model parameters M'_{local_i} are divided into shares using Shamir's Secret Sharing. The facility computes:

$$\text{ShareSplit}(M'_{local_i}, n) \rightarrow \{s_{i1}, s_{i2}, ..., s_{in}\}$$

where n is the number of fog nodes. Each share $s_{ij}$is transmitted securely to the validator committee for verification. Once validated, the committee members perform the broadcast of approved shares to their respective fog nodes as follows:

$$Broadcast(Committee, s_{ij}) \rightarrow FogNode_j$$

This ensures that each fog node j receives the correct and authenticated share s_{ij} of the differentially private local model. The fog nodes later use these shares in the partial aggregation phase.

The process of local model training, perturbation with differential privacy, and splitting into secret shares can be illustrated by Algorithm 4 and Algorithm 5.

2.4.4 Validation and Committee Mechanism

After each facility generates shares of its differentially private local model, these shares are sent to the validator committee for integrity verification and authentication before being broadcast to fog nodes. The validator committee ensures Byzantine fault tolerance, Sybil resistance, and trust in the federation. Proof-of-Work for Facility Validation Before any facility can submit its shares, it must solve a computational puzzle to prevent spam and Sybil attacks. The Proof-of-Work (PoW) mechanism is defined as:

Algorithm 4 Local Training with Differential Privacy

Input: Initial model M_{init}, local dataset d, learning rate η, number of epochs E, noise scale σ
Output: Differentially private local model M_{local}^{DP}

for $epoch = 1$ to E **do**
    Compute gradient: $\nabla F_i(M_{init})$ using local dataset d
    Add DP noise: $\tilde{\nabla} \leftarrow \nabla F_i(M_{init}) + \mathcal{N}(0, \sigma^2)$
    Update model: $M_{local}^{DP} \leftarrow M_{init} - \eta\tilde{\nabla}$
end for

Algorithm 5 Secret Sharing and Broadcast to Validator Committee

Input: Local model M_{local}^{DP}, number of fog nodes n
Output: Secret shares $S_1, S_2, ..., S_n$

Divide M_{local}^{DP} into n shares using Shamir's Secret Sharing
for each share S_i **do**
    Send S_i to validator committee for verification
end for
Broadcast approved shares to fog nodes

$$H(N \parallel ID_{facility}) < T$$

where:

- H is a cryptographic hash function,
- N is a nonce,
- $ID_{facility}$ is the facility's identity,
- T is the system-defined difficulty threshold.

Only if the hash output is below the target T, the facility's request is accepted by the committee. This ensures that each facility invests computational effort before being able to contribute.

Digital Signature Authentication

Each facility signs its shares before sending them to the validator committee:

$$Sign(SK_{facility}, s_{ij}) \rightarrow Sig_i$$

where $SK_{facility}$ is the facility's signing key. The committee verifies:

$$Verify(PK_{facility}, s_{ij}, Sig_i) = \text{True}$$

If the verification fails, the share is discarded. This guarantees authenticity and prevents tampering.

Committee Consensus Voting

Once authenticated, the committee validates whether the share is consistent and well-formed. Each committee member C_k casts a binary vote (1 = valid, 0 = invalid). The final decision is determined by majority consensus:

$$Decision(s_{ij}) = \begin{cases} 1, & \text{if } \sum_{k=1}^{m} Vote_k(s_{ij}) \geq \frac{m}{2} + 1 \\ 0, & \text{otherwise} \end{cases}$$

where m is the number of committee members.
Only if the majority votes "valid," the share is approved.

Secure Broadcast

Approved shares are then broadcast to their respective fog nodes:

$$Broadcast(Committee, s_{ij}) \rightarrow FogNode_j$$

Each broadcast message is signed by the committee to ensure end-to-end integrity:

$$Sign(SK_{Committee}, s_{ij}) \rightarrow Sig_{Committee}$$

The fog node accepts a share only if:

$$Verify(PK_{Committee}, s_{ij}, Sig_{Committee}) = \text{True}$$

This ensures that fog nodes receive only verified and committee-approved contributions.

The validator committee's verification, consensus voting, and secure broadcast of approved shares are captured in Algorithm 6.

2.4.5 Fog Node Aggregation

After validation, the fog nodes assume the role of regional aggregators. Their purpose is to combine verified local model updates received from medical facilities in their region. This intermediate aggregation reduces communication overhead, strengthens fault tolerance, and ensures scalability before sending the results to the leader server for global aggregation.

Local Model Fragments Collection

Algorithm 6 Validator Committee Verification

Input: Secret shares $S_1, ..., S_n$, digital signatures of facilities
Output: Verified shares for aggregation

1: **for** each received share S_i **do**
2: Verify digital signature of facility
3: Check share integrity and consistency
4: **end for**
5: Apply consensus voting among committee members to approve or reject shares
6: Forward verified shares to respective fog nodes

Each medical facility i sends its validated model shares $S_{i,j}$ to the fog node j. Here, n denotes the total number of fog nodes, and $S_{i,j}$ represents the fragment of model parameters split for fog node j. The collection process can be expressed as:

$$S_j = \{S_{1,j}, S_{2,j}, \ldots, S_{m,j}\}$$

where m is the number of medical facilities assigned to fog node j.

Partial Aggregation Formula

Fog nodes apply a Federated Averaging (FedAvg) scheme to combine the shares received. The partial aggregated model at fog node j is defined as:

$$M_{fog_j} = \frac{1}{m_j} \sum_{i=1}^{m_j} S_{i,j}$$

where m_j is the number of facilities contributing to fog node j. This ensures that each fog node outputs a compact intermediate model update.

Broadcast to Leader Server

Once the aggregation is complete, each fog node broadcasts its partial aggregated model to the leader server. The broadcast can be formalized as:

$$Broadcast(M_{fog_j}) \rightarrow Leader$$

The leader server then receives all partial models from the n fog nodes:

$$M_{fog} = \{M_{fog_1}, M_{fog_2}, \ldots, M_{fog_n}\}$$

These results will serve as input for the final global aggregation step.

Security and Authentication

Before sending, fog nodes attach digital signatures to their partial models. This prevents tampering during transmission and ensures integrity. The leader server verifies each signature before proceeding with global aggregation:

$$Verify(Sign_{fog_j}(M_{fog_j})) = True \quad \forall j \in \{1, \ldots, n\}$$

Only authenticated and validated partial models are accepted.

The partial aggregation by fog nodes, including collection, FedAvg computation, signing, and forwarding to the leader server, is described in Algorithm 7.

Algorithm 7 Fog Node Partial Aggregation

Input: Verified shares $S_{1..n}$ from facilities in the fog region
Output: Partially aggregated model M_{fog}

1: Apply FedAvg on received shares: $M_{fog} \leftarrow \frac{1}{n} \sum_{i=1}^{n} S_i$
2: Send M_{fog} to leader server for global aggregation

2.5 Leader Server Aggregation

The leader server, randomly selected from the pool of fog nodes, is responsible for performing the global aggregation. Its primary role is to collect the partial models from fog nodes, verify their authenticity, and compute the updated global model.

2.5.1 Input Retrieval

The leader server retrieves all partial models M_{fog_j} from the fog nodes according to its assigned index $Index_{Leader}$:

$$M_{fog} = \{M_{fog_1}, M_{fog_2}, \ldots, M_{fog_n}\}$$

2.5.2 Verification of Authenticity

Each received model is verified using digital signatures to ensure authenticity and integrity:

$$Verify(Sign_{fog_j}(M_{fog_j})) = True \quad \forall j \in \{1, \ldots, n\}$$

2.5.3 Global Aggregation

The leader server performs the global aggregation strictly as a summation of all valid fog node contributions:

$$M_{global} = \sum_{j=1}^{n} M_{fog_j}$$

This produces the unified global model, which represents the collective knowledge of all participants.

2.5.4 Global Model Redistribution

After the leader server computes the global aggregation:

$$M_{global} = \sum_{j=1}^{n} M_{fog_j}$$

it distributes the resulting global model to all registered medical facilities. The broadcast is defined as:

$$Broadcast = \{Index_{user1}||M_{global},\ Index_{user2}||M_{global},\ \ldots,\ Index_{usern}||M_{global}\}$$

Each facility receives the aggregated global model M_{global} according to its index. Once received, each facility initializes local training. The model is trained for a specified number of epochs E using the facility's local dataset D_i. The local update rule is:

$$M_{local_i} = M_{global} - \eta \nabla F_i(M_{global})$$

where:

- η is the learning rate,
- $F_i(x)$ is the local loss function for facility i,
- ∇ is the gradient operator.

After training, each user divides the updated model into secret shares according to the number of fog nodes n, then sends them for committee validation and subsequent partial aggregation. This broadcast–train–share cycle repeats until the global model M_{global} converges to its final state. The leader server's retrieval of partial models, verification, global aggregation, and redistribution of the global model are illustrated in Algorithm 8.

Algorithm 8 Global Aggregation at Leader Server

Input: Aggregated models from all fog nodes $M_{fog_1}, ..., M_{fog_n}$
Output: Global model M_{global}

1: Compute summation: $M_{global} \leftarrow \sum_{j=1}^{n} M_{fog_j}$
2: Broadcast M_{global} to all participants
3: Participants update local models: $M_{local} \leftarrow M_{local} - \eta \nabla F_k(M_{global})$