International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

ENHANCING THE PRIVACY AND INTEGRITY OF FEDERATED LEARNING MODELS USING HYBRID HOMOMORPHIC ENCRYPTION

Zainab Jawad Al-Abedi[1], Ali Taei Zadeh[2]

[1,2] Faculty of Technical and Engineering, University of Qom, Qom, Iran

E-mail address: Zainabsomo@gmail.com, a.taeizadeh@qom.ac.ir

Abstract

Federated learning (FL), is a recent concept to decentralize the model training process from raw data sharing. The traditional setup of federated learning, however, is still prone to privacy attacks during model aggregation and transmission stage. To answer this issue, we propose a privacy-preserved federated learning framework integrating Hybrid Homomorphic Encryption with secure aggregation and distributed key management. We efficiently deploy Partial Homomorphic Encryption for the linear operations. We utilize Fully Homomorphic Encryption for non-linear computations to achieve end-to-end encrypted learning. The framework features a modular design consisting of clients, edge aggregation servers, and a global server. Each component works on encrypted data. We design the algorithms for local model encryption, encrypted aggregation, and global model updates such that at no time will sensitive information be exposed during the entire learning process. Experiments with the MNIST dataset show that the proposed method has higher accuracy and lower training loss than the baseline FedAvg algorithm with only moderate time overhead. We also analyze the encryption and decryption performance with varying key lengths to demonstrate the scalability and flexibility of the system. We evaluate our framework using the MNIST dataset under both i.i.d. and non-i.i.d. settings. The performance is benchmarked against the baseline FedAvg algorithm across multiple metrics including classification accuracy, training loss, encryption/decryption time, and total communication overhead. Results show that our method consistently achieves higher accuracy and lower training loss while maintaining a moderate computational cost. This demonstrates its superiority in both privacy preservation and learning performance compared to conventional federated approaches.

Keywords: Federated Learning; Privacy Preservation; Homomorphic Encryption; Hybrid Encryption; Secure Aggregation; Data Confidentiality; Model Integrity; Edge Computing; Trust Chain; Cryptographic Overhead

1. Introduction

The emergence of edge computing paradigms like smartphones and Internet of Things sensors has led to a significant increase in the volume of data generated by distributed devices. The need for distributed approaches to machine learning is highlighted by this explosion in decentralized data creation. This need is met by federated learning (FL), which allows several users to work together to train a common model without sharing raw data with a central server. FL is a promising solution for privacy-sensitive industries like healthcare and finance because of its decentralized architecture, which naturally improves data privacy and lowers communication costs [1,2]. Federated Learning (FL) is a new paradigm that permits a large number of participants to jointly train a common model without transferring raw data to a central server [3]. The 'decentralization' in this approach serves to address the data privacy concerns and to save on communication costs [4]. On the contrary, conventional FL systems

Received: August 11, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

are always subjected to various security threats that include data leakage threats during the model update transmission, malicious aggregations, and inference attacks that derive clients' private data from shared model parameters [5].

To counter this, more recent research has strived towards the realization of embedding cryptographic tools at different stages of the FL pipeline [6]. In this context, homomorphic encryption has been a big favorite since it allows computations to be carried on encrypted data [7]. With homomorphic encryption, the model updates remain confidential during aggregation, such that neither the central server nor intermediary nodes have access to sensitive client information [8]. However, the widespread implementation of fully homomorphic encryption throughout the entire training process usually means high computational costs [9]. Therefore, hybrid solutions combining partial homomorphic encryption for linear operations and fully homomorphic encryption for non-linear components strike a balance between security and performance [10].

Not only data confidentiality, but integrity and auditability over the federated learning process are also essential [11]. For this end, a trust chain-a blockchain-like architecture that stores events of data processing in a tamper proof and transparent way-can improve FL system reliability. Mechanisms like this are especially useful in the circumstances of little trust between participants or when legislation requires verification of such transaction histories [12].

This paper introduces an innovative FL framework designed to prioritize privacy by combining hybrid homomorphic encryption with distributed trust mechanisms. The framework features separate modules for local training, generating encrypted model updates, secure aggregation at edge nodes, and refining the global model. To ensure complete data confidentiality and protect against single points of failure, a distributed key management architecture is employed. The main goal of this research is to create a scalable, privacy-preserving federated learning system that strikes a balance between computational efficiency and robust cryptographic protections. With that in mind, the study is driven by the following research questions: 1. How can we effectively integrate hybrid homomorphic encryption into FL to lower computation costs while maintaining privacy? 2. How does the proposed system stack up against existing FL frameworks like FedAvg in terms of accuracy, training loss, and encryption overhead? 3. Can a distributed key management approach boost the trust and security of federated learning systems without creating centralized vulnerabilities?

A distributed key management architecture cuts all single points of failure, and all computations are performed on encrypted data so that end-to-end confidentiality is preserved.

We evaluate our approach over the MNIST dataset under both i.i.d. (independent and identically distributed) and non-i.i.d. settings and compare it with the baseline FedAvg. Results show that our framework preserves or enhances model accuracy while very moderately adding cryptographic overhead. The proposed system is scalable, robust against data heterogeneity, and applicable for deploying in real-world applications that are sensitive from the privacy point of view, like healthcare, finance, and smart environments.

In contrast to other secure aggregation schemes, like those found in [10], our hybrid approach combines Fully Homomorphic Encryption (FHE) for non-linear updates and Partial Homomorphic Encryption (PHE) for linear operations in a modular federated setting. The majority of earlier works only use trusted execution environments or a single homomorphic scheme.

Received: August 11, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

In addition, unlike centralized key servers found in many current systems, we present a distributed key management architecture that removes single points of failure. Secure key refresh protocols, segmented authority, and decentralized key generation are all included in our key management component.

In conclusion, this work's primary contributions are:

1. A hybrid encryption framework that uses PHE and FHE selectively to strike a balance between security and computational efficiency.
2. A new distributed key management system that improves robustness and trust in federated settings.
3. A secure aggregation method that doesn't require intermediate decryption and works only with encrypted data.
4. In comparison to FedAvg, empirical evaluation under both i.i.d. and non-i.i.d. conditions demonstrates superior accuracy and acceptable overhead.

2. Related Works

Federated Learning (FL) has very recently been on the rise as one of the contenders in solving privacy-related concerns in decentralized data environments by allowing a collaborative way of model training without the exposure of raw data [7,13]. Nevertheless, attacks on inference, model poisoning, and privacy leakage go invariably hand in hand with the FL system's transfer of model updates. This has led most recent investigations into embedding cryptography, particularly Homomorphic Encryption (HE), in the FL pipeline [14]. Several recent studies have been dedicated to strengthening FL with homomorphic encryption to give privacy to model updates. For instance, [13] demonstrate that computations on ciphertexts in an FL scheme using Fully Homomorphic Encryption (FHE) for IoT applications can maintain both privacy and performance of the model at the expense of latency. [14] presented a similar idea in an efficient FL model using single-key homomorphic encryption in combination with elliptic curve cryptography and trusted execution environment, which was able to withstand collusion and practical for under-resourced settings.

Blockchain and FL formed an ever-exploring concept to ensure transparency and integrity. [10] built the foundation of homomorphic encryption with a blockchain-and-reputation-based mechanism to allow encrypted model aggregation and incentivization for nodes' participation. [15] extended this concept onto IoT supply chains, demonstrating decentralized integrity verification through blockchain while ensuring homomorphic encryption for data during transmission and processing. [16], however, proposed yet another algorithm called the MHEF-AHDS. This was a Modified Homomorphic Encryption Federated-based Adaptive Hybrid Dandelion Search algorithm. With respect to IoT data, their approach further gained ground on protecting sensitive data with respect to encryption and optimization as well as blockchain to secure accuracy against federated learning. [17] proposed a privacy-preserving FL framework using HE combined with a trust chain structure and edge computing for scalable and efficient deployment, thus obtaining performance beyond FedAvg models. Other notable works target intrusion detection via federated learning. [18] proposed an Enhanced Ghost_BiNet architecture with homomorphic encryption to boost detection accuracy for FL-based network security systems. Their method incorporated deep learning with optimization to enhance performance across varied security metrics. With resistance against adversarial model poisoning attacks, [9] introduced a different privacy-preserving FL model employing

additive homomorphic encryption and gradient robust through Gaussian Mixture Models and Mahalanobis distance, with high accuracy against malicious encrypted updates.

The work of [7] provides a thorough overview of HE optimization for privacy-preserving FL from the perspectives of architecture and efficiency, detailing the trade-offs that exist among security, latency, and scalability. Their classification into hardware, algorithmic, and hybrid approaches represents a solid reference for developing efficient yet privacy-conscious FL systems.

In conclusion, much work has advanced the integration of federated learning with encryption, blockchain, and edge computing. Yet, much of this work looks at privacy purely at the expense of computation efficiency, or integrity without strong guarantees on end-to-end encryption. Further, they mainly depend on static encryption settings or a centralized key management system; this leaves opportunities to enhance areas such as hybrid encryption, distributed trust management, and scalable aggregation methods, which is the gap our framework intends to fill.

Table 1. Comparison of Security Methods in Federated Learning Frameworks

Study	Encryption Method	Key Management	Aggregation	Scalability	Privacy Guarantee
[10]	PHE + Blockchain	Centralized	Encrypted + Reputation	Moderate	Strong
[13]	FHE	Centralized	Encrypted	Low	Very Strong
[14]	Single-key HE + TEE	TEE-managed	Encrypted	Moderate	Strong
[15]	Blockchain + HE	Centralized	Encrypted	Moderate	Strong
[17]	HE + Trust Chain	Partial Trust-based	Encrypted + Trust Chain	High	Very Strong
Proposed Method	Hybrid HE (PHE+FHE) + Distributed Key Mgmt	Fully Distributed	Fully Encrypted + Secure Aggregation	High	Very Strong

3. Method

In this paper, we develop a federated learning framework augmented by privacy through the combination of hybrid homomorphic encryption (HHE) with trusted aggregation techniques, to ensure data privacy, model integrity, and computation efficiency according to the design. It leverages partially and fully homomorphic encryption for the overhead security trade-off and incorporates a distributed key management scheme to eliminate single points of failure.

The specialized algorithms and workflows for clients, aggregation servers, and key management authority are as follows: local encryption, secure aggregation of models, and a global model update steps. Following this, the description includes system architecture, threat model, and process details.

3.1 System Model

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

The proposed privacy-preserving federated learning framework using hybrid homomorphic encryption (HHE) constitutes a decentralized network consisting of four main components, namely Clients (Participants), Aggregation Servers, Global Server, and Key Management Authority (KMA). The purpose of these components is to maintain confidentiality, integrity, and efficiency in the federated learning process performing the model aggregation and data transmission steps under restricted security. The system architecture for privacy-preserving federated learning with respect to hybrid homomorphic encryption approaches is presented in Figure 1.

1. Clients (Participants)

The clients are the devices or entities that make the local model updates based on their private datasets. Each client performs local training on its own data and computes model update variables (such as gradients, or parameters) by iterating through the input data. The major feature attributed to these entities is the ability of keeping the data private using encryption techniques, mainly after the model updates are encrypted and sent to the aggregation servers.

- **Local Model Training:** The client owns and trains the local model based on local data, so no actual raw data needs to be shared. The training techniques being used include techniques such as gradient descent.
- **Encryption:** Updates to the local model are encrypted through a hybrid encryption method whereby some parts of the data are encrypted using a partial homomorphic encryption technique such as Paillier for simple operations like addition and weighted sums, whereas the more complex portions of the updates are encrypted through leveled fully homomorphic encryption techniques such as BGV or CKKS.

2. Aggregation Servers

The most important functions of aggregation servers include securely aggregating the model updates sent from many clients. These servers receive encrypted model updates and aggregate them in encrypted form without ever decrypting the data, thereby ensuring confidentiality of sensitive information.

- **Secure Aggregation:** The aggregation process is performed on the encrypted model updates using homomorphic encryption. Specifically:
  - **Partial Homomorphic Encryption:** The summation of gradients, which is a linear operation, can be performed using the Paillier encryption scheme. This allows aggregation of the encrypted updates for the model while maintaining data privacy.
  - **Fully Homomorphic Encryption:** In case of a more complicated transformation or non-linear operations, fully homomorphic encryption, such as CKKS or BGV, is used. With this, all operations applied to the data are performed on an encrypted state.
- **No Data Decryption:** Along the way, servers downloading new models cannot decrypt them, preventing exposure of private information.

3. Global Server

The global server collects the aggregated encrypted model updates from the aggregation servers and performs the final update on the global model; it can be centralized or decentralized, depending on the system architecture. After aggregation, the updated global model is sent back to the clients from the global server for further local updates.

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- **Global Model Update:** At the server, the final aggregation on the encrypted results takes place. Subsequently, the updated model is sent back to the clients for another round of local training.
- **Privacy Preservation:** The global server does not deal with any raw data or decrypted model updates, ensuring the privacy of participants' data.

4. Key Management Authority (KMA)

The KMA is a trusted authority that creates and distributes cryptographic keys used for encryption and decryption. The KMA generates key pairs for each participant, aggregation server, and global server. The distribution of these keys is made in such a way that every entity can encrypt and decrypt the data necessary for its operations without having to share the private keys.

- **Key Generation and Distribution:** The KMA produces public and private key pairs for all participating entities in a hybrid homomorphic encryption scheme whose public keys are distributed among clients and servers while the private keys remain securely stored with the relevant entities.
- **Key Management Security:** The KMA ensures secure and controlled access to keys. This is done to avoid unauthorized use and to support the protection of the cryptographic process.

5. Communication and Workflow

The overall workflow of the system is as follows:

1. **Key Distribution:** Public keys are handed out by the KMA to clients and aggregators. entities such as clients and servers maintain or restrict access to the private keys.
2. **Local Training and Encryption:** The clients carry out local model training based on their data and then encrypt the updates using public key encryption. Such encrypted updates will be sent to the aggregation servers.
3. **Secure Aggregation:** The aggregation servers do encryption on the aggregation for the updates of the clients in such a way that it assures at no time does any sensitive data leave its secure environment.
4. **Global Model Update:** Global Servers accumulate all the updates, perform a global update in encrypted form and send the updated model back to the clients for the next round of training.
5. **Decryption:** Clients decrypt the aggregate model using their private keys and update their local models accordingly.

6. Security Considerations

The suggested system guarantees complete secrecy of data and also integrity of the model during federated learning processes. All operations are performed on homomorphically encrypted data without any leakage of sensitive information. The Key Management Authority distributes and manages the cryptographic keys securely, which minimizes unauthorized access risk. Additionally, the semi-honest model assumption protects such aggregation servers and the global server from malicious alterations of model updates, hence ensuring the integrity of the system.

Received: August 11, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

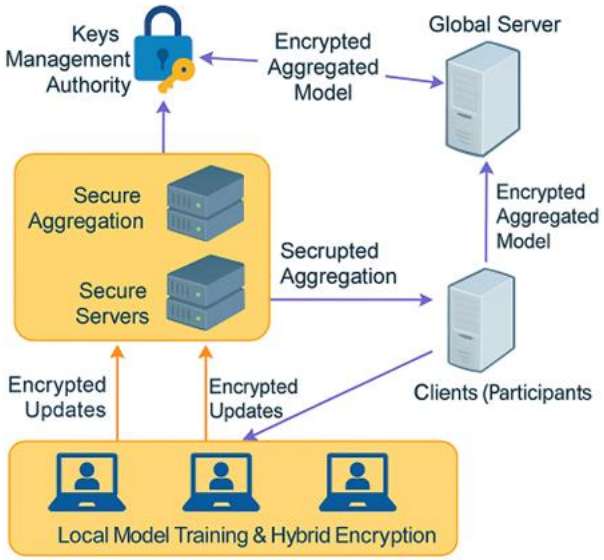

Figure 1. Federated learning privacy protection system model.

With the current system model, the above-mentioned hybrid homomorphic encryption with secure aggregation has provided an efficient solution to privacy-preserving federated learning, guaranteeing data confidentiality and model integrity during the entire learning process.

3.1.1 Threat Model and Design Goals

Threat Model

With that, let's turn our attention to an introduction to hybrid homomorphic encryption and its role in securing federated settings, especially in connection with privacy-preserving federated learning systems. Some important threats addressed in the threat model of the proposed framework include:

1. **Eavesdropping:** External attackers might intercept encrypted communications, but encryption ensures data privacy during transmission.
2. **Data Leakage during Aggregation:** Aggregation servers could infer sensitive data from model updates. Homomorphic encryption prevents data leakage during aggregation.
3. **Model Inference Attacks:** The global server should not access raw data from aggregated models. All operations are conducted on encrypted data to prevent this.
4. **Key Management Risks:** The KMA could be a target. Distributed key management reduces reliance on a single trusted entity.
5. **Malicious Clients:** There may exist malicious participants trying to manipulate the system, but global model's integrity remains protected by encryption.

Design Goals

The system’s design goals are:

1. **Data Privacy:** Ensuring data remains encrypted throughout the learning process.
2. **Model Integrity:** Guaranteeing the accuracy of the aggregated model without tampering.

Received: August 11, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

3. **Efficiency and Scalability:** Balancing security with computational efficiency for large-scale systems.
4. **Secure Aggregation:** Preventing unauthorized access to data during aggregation.
5. **Key Management:** Decentralized and distributed key management for the avoidance of potential single-point-of-failure situations.
6. **Adaptability:** Ensuring the system operates efficiently in heterogeneous environments.

These goals aim to protect privacy, maintain model integrity, and ensure secure and efficient computations in federated learning.

3.1.2 System Workflow

The methodology of the suggested privacy-preserving federated learning system equipped with hybrid homomorphic encryption (HHE), is organized into a set of structured steps for achieving data privacy, model integrity, and efficient aggregation. The workflow is that between the Clients, Aggregation Servers, Global Server, and the KMA. Below is the step-by-step procedure:

1. **System Initialization and Key Distribution:**

- The KMA creates public-private key pairings for every participant, aggregation server, and the global server individual client.
- Public keys are distributed among the clientele and server securely, while private keys are safely stored with each entity.

2. **Task Initialization:**

- The initiation of the federated learning task by requester ends with sending task details such as model parameters, data requirements, and target accuracy to the global server.
- After that, the task along with the public keys is passed on to the clients for local training by the global server.

3. **Local Model Training and Encryption:**

- Each consumer utilizes its in-house data to build local model training. From there, the consumer encrypts the model updates using the KMA public key provided.
- These encrypted model updates are then uploaded to the aggregation server.

4. **Secure Aggregation at Aggregation Servers:**

- The clients send encrypted updates of the model to an aggregation server. The server aggregates the encrypted updates using homomorphic encryption.
- No decryption occurs, so sensitive data is always protected. The aggregated model still in the encrypted format is then sent to the global server for further processing.

5. **Global Model Update:**

- The aggregation servers send aggregated encrypted model updates to the global server.
- The server does global aggregation, using the encrypted data, for the purpose of updating the global model.
- The updated global model is sent back to clients for the next round of local training.

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

6. **Decryption and Local Model Update:**

- Each client decrypts the key update through their own private key.
- The client will therefore perform model updates to its local models by considering the global model after decrypting and then subsequently prepare to run another training round.

7. **Repetition of the Process:**

- This process repeats for multiple rounds until the model converges and meets the desired performance.

All the way down to model updates, every possible data is encrypted during transmission and aggregation-the core elements of robust privacy protection. Efficiency in computing blended with a strong security aspect constitutes hybrid homomorphic encryption-the essence of scaling and functional performance for large distrubuted environments. The global workflow of the privacy-preserving federated learning system is illustrated in Figure 2. Local model updates are secured with hybrid homomorphic encryption during transmission and aggregation.

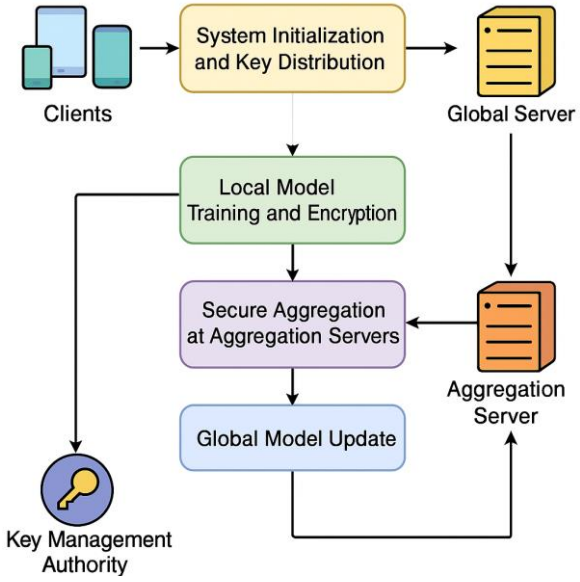

Figure 2. Workflow of privacy-preserving federated learning system using hybrid homomorphic encryption.

3.2 Algorithm Description

This section describes the proposed algorithm for enhancing the privacy and integrity of federated learning models based on Hybrid Homomorphic Encryption (HHE). The workflow involves local training, hybrid encryption, secure aggregation, and global model update.

Step 1: System Initialization

Before it is further delineated down, the Key Management Authority (KMA) starts generating key pairs for their members as a matter of course.

- Public and private keys for Partial Homomorphic Encryption (PHE), such as the Paillier scheme.

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- Public and private keys for Fully Homomorphic Encryption (FHE), such as the CKKS or BGV schemes.

The key distribution is conducted as follows:

- Public keys $_{fhe}pk_{phe}, pk$ are securely distributed to clients and aggregation servers.
- Private keys $_{fhe}sk_{phe}, sk$ are securely retained by each corresponding entity.

Step 2: Local Model Training and Encryption

Each client i trains its local model on private dataset D_i to obtain the model parameters w_i. The encryption: process is as follows

- Linear components (e.g., gradients) are encrypted using PHE

$$Enc_{phupk}(w_i) = E_{phe}(w_i)$$

- Non-linear or complex components are encrypted using FHE

$$Enc_{fhepk}(w_i') = E_{fhe}(w_i')$$

The encrypted local updates $\{E_{phe}(w_i), E_{fhe}(w_i')\}$ are transmitted to the aggregation servers

Step 3: Secure Aggregation at Aggregation Servers

Aggregation servers perform secure aggregation without decrypting the data:

For linear components:

$$E_{phe}(w_N) \oplus \cdots \oplus E_{phe}(w_2) \oplus E_{phe}(w_1) = E_{phe}\left(\sum_{i=1}^{N} w_i\right)$$

For non-linear components:

- Calculation is made on the cipher texts per FHE properties.
- Aggregated encrypted responses are sent to the global server.

Step 4: Global Model Update at Global Server

Afterward, the consolidated encrypted updates are sent to the global server from aggregation servers which results from the simulation process conducted by some of the systems:

- Aggregation at a global level within ciphertext could be done without initiating any decryption process.
- On the contrary, a partial decryption with distributed keys can be performed for recovering the aggregated model update.

The global model is updated as follows:

$$\eta\left(\sum_{i=1}^{N} \Delta w_i\right) + {}_{g}^{(t)}w = {}_{g}^{(t+1)}w$$

Received: August 11, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

where η is the learning rate.

Step 5: Model Distribution to Clients

- The centralized model is downloaded to the clients now.
- Each client decrypts the global model with its private key and updates its local model for the next training round.

Step 6: Iterative Training

Steps 2 through 5 shall be repeated before any communication round has been conducted up to the time that the global model discovers its convergence. Throughout the succession[offset]:

- Client data remains private and encrypted.
- Aggregation servers and the global server never access raw data.
- Key recovery and re-encryption mechanisms are accessible to deal with faults or security breaches.

The proposed method illustrated in Figure 3 encompasses the procedure in a flowchart representation, comprising a series of sequential steps: local training, encryption, secure aggregation, and update of the global model, altogether under the hybrid federated learning framework.

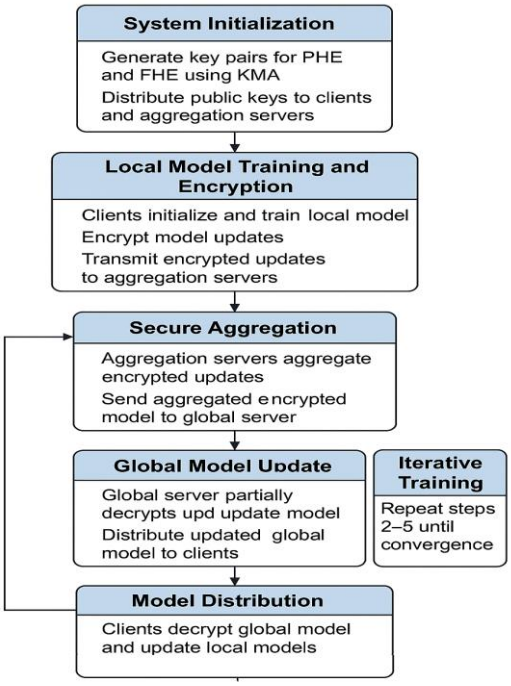

Figure 3. Flowchart of the Proposed Method

3.4 Security Analysis

This part assesses various security properties of proposed privacy covering federated learning framework using Hybrid Homomorphic Encryption (HHE). The major aim is guaranteeing

data confidentiality, integrity of model, and secure computation across federated learning for the intended operation it is supposed to do.

3.4.1 Data Confidentiality

End-to-End Encryption: Following the complete encryption of local model updates on the client side, transmission to the aggregation servers takes place. Updates to the linear components are encrypted with PHE and the non-linear components with FHE.

Homomorphic Computation: Carried out operations of aggregation can be performed without the need of decryption for encrypted data. Thus, aggregation servers cannot access nor infer any sensitive information from the updates from the model.

Protection against Eavesdropping: An attacker watching an encrypted connection does not necessarily indicate a threat against the security of users.

3.4.2 Model Integrity

Integrity of Aggregation: The homomorphic attributes thus guarantee that these aggregation servers will not be able to alter the updates without any detection. Any modification of those encrypted values would actually corrupt the resulting model and would be caught in the decryption and validation steps.

Secure Global Model Update: The global server does aggregation purely on the authenticated and encrypted data, which ensures that the global model gets updated accurately without any unauthorized tampering.

3.4.3 Threat Model Resilience

Semi-Honest Servers: That the aggregation servers and the global server are semi-honest here implies that they do follow the protocol in earnest but may attempt to gain some additional knowledge. Again, the HHE framework is designed in such a way that even the semi-honest adversaries cannot acquire any information of any value.

Malicious Clients: Even though the clients might try to upload an update that was malformed or adversarial, validation mechanisms at the aggregation stage, coupled with encryption, would detect and filter those anomalies.

Key Management Security: KMA is an authorized agency to generate and distribute encryption keys in a secured manner. Even if a limited subset of keys gets compromised, it could avoid large-scale compromises through distributed key refreshment protocols.

3.4.4 Defense Against Common Attacks

Model Inference Attacks: Encryption thwarts application servers and the global server from piecing together client datasets using model updates.

Collusion Attacks: Even with collaboration among several aggregation servers, clearing model updates is still not possible unless they have the private keys of the clients, which remain safely protected.

Replay and Injection Attacks: Out-of-date bulletin-starred messages or forgeries cannot be injected into the consolidation mechanism in the timestamping and nonce-based security systems.

3.4.5 Formal Security Guarantees

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Let $\mathcal{A}$ be a probabilistic polynomial-time (PPT) adversary, and ${}_{\mathcal{A}}Adv$ denote the advantage of $\mathcal{A}$ in breaking the confidentiality of a client's local model. Assuming the semantic security of the PHE and FHE schemes :(under standard cryptographic assumptions like IND-CPA security), we have

$$\epsilon \geq {}_{A}Adv$$

Here, ϵ must be a negligible function of some security parameter. The proposed framework, thus, achieves strong semantic security with universally recognized security assumptions.

4. Experiments

4.1. Dataset

To evaluate the privacy-preserving federated learning framework proposed in this study, we use the MNIST dataset, which is popularly known. The MNIST dataset contains 70,000 grayscale images of handwritten digits (0-9), which are divided into 60,000 training images and 10,000 testing images. Each image in the dataset has the dimensions 28×28 pixels, with gray shades represented by pixel values ranging from 0 to 255. Thus, all these images are passed as a 1D array of size 784 (28×28) to the input for the machine learning models.

In a federated learning scenario, the entire dataset is distributed amongst *N* clients, each having some part of the data. In a non-i.i.d. (non-independent and identically distributed) setup, the data would be distributed in such a way that each client would only have access to images belonging to two distinct digit classes, which is probably a better representation of data distribution since each device may have a biased dataset. This distribution may pose difficulties for federated learning as regards model convergence and performance [19, 20].

This study presents the preprocessing steps needed to prepare the dataset for training purposes. For pixel value normalization, the raw pixel values of the images have been divided by 255, which scales the values to the range between 0 and 1 and consequently increases the training efficiency. The second step involves flattening the image to a one-dimensional array of size 784 to serve as the input to the machine learning models. Finally, the class labels are encoded as integers from 0 to 9 representing the corresponding digits in the image.

Federated learning can protect the data privacy since the raw data will always stay locally on each client and will never be shared or transmitted to other clients or the aggregation server. Each client calculates local model updates based on its private data and shares only these encrypted model updates with the central server. This guarantees that sensitive information remains confidential throughout the entire learning process [21].

Furthermore, to test the robustness of the federated learning framework with respect to various data distributions, the MNIST dataset is distributed in different configurations across the clients. The experiment examines both i.i.d. and non-i.i.d. data splits providing insights about the performance of the system in different real-world scenarios in which clients might have different amounts or types of data distributions.

4.2 Experimental Setup

The proposed privacy-enhanced federated learning framework based on Hybrid Homomorphic Encryption (HHE) is assessed in a three-layer architecture of the client layer,

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

aggregation layer, and global server layer. This simulation is meant to imitate a realistic federated learning environment wherein local training, encryption, aggregation, and global model updates are performed in a decentralized yet secure manner.

In the client layer, each participant independently trains the local model using a batch of data from the MNIST dataset. After the completion of the training, participants encrypt their respective local model parameters through hybrid homomorphic encryption based on Partial Homomorphic Encryption (PHE) for linear operations and Fully Homomorphic Encryption (FHE) for non-linear computations. After encryption, the updates are sent to the aggregation layer.

Yet another layer of security mimicking many edge nodes receiving encrypted updates from their assigned clients: these edge nodes perform secure aggregation without decrypting the data, using the encryption schemes' homomorphic properties. The aggregated results in encrypted form are forwarded to a global server.

The global server layer is the heart of the server where aggregation occurs among all encrypted data coming from all edge nodes. After the aggregation is completed, the updated encrypted model shall then be sent to clients again. The clients decrypt it with their private keys and later go on to the next training round.

All the simulations have been carried out in Python whereas the deep learning library was PyTorch for training and model handling and the phe library provided functionality for the Paillier homomorphic encryption. For components of FHE, the library TenSEAL was used to simulate CKKS operations on those non-linear parts of the model.

To make realistic performance analysis, the following system parameters were defined:

Table 1. Simulation parameters

Parameter Name	Default Value
Global Aggregation Rounds	20
Edge Aggregation Rounds	5
Local Training Iterations	20
Batch Size	20
Learning Rate (η)	0.01
Learning Rate Decay Factor	0.995
SGD Momentum	0.9
Encryption Key Length (Paillier)	512 bits
Number of Clients	50
Number of Edge Servers	5
Data Distribution	Non-IID (two classes per client)

Received: August 11, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

4.3 Evaluation Metrics

In order to assess thoroughly the performance, privacy, and efficiency of the proposed HHE-enhanced federated learning system, events are triggered for assessment based on four fundamental metrics, namely: classification accuracy, training loss, time overhead and encryption/decryption time. Definitions and formal expressions for these metrics are provided below:

Classification Accuracy

This corresponds to the predictive performance of the model on a labeled test data set. It can be defined in terms of the ratio of the number of correctly classified instances and the total number of instances. If N is the total number of test samples, then accuracy can be given in the following form:

$$\text{Accuracy } = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(\hat{y}_i = y_i)$$

where $\hat{y}_i$ is the predicted label for the i th sample, y_i is the true label, and $\mathbb{I}$ ($\hat{y}_i = y_i$) is an indicator function that equals 1 if the prediction is correct (i.e. $\hat{y}_i = y_i$) and 0 otherwise. A higher accuracy (closer to 1.0 or 100%) indicates better classification performance of the global model.

Training Loss

Training loss is the measure of how much the error between the predicted values and true labels is during training. This loss tells us how much the model is learning and converging. We are using cross-entropy loss function in our classification setting. The average cross-entropy loss is given as below for N training samples, having C possible classes:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log\left(\hat{p}_{i,c}\right)$$

where $y_{i,c}$ is the one-hot ground truth indicator (1 if sample i belongs to class c, otherwise 0) and $\hat{p}_{i,c}$ is the predicted probability of sample i belonging to class c. Higher value of loss indicates larger difference between prediction and target which means worse fit and convergence of the model. So we monitor training loss across the rounds for the benchmark experiments to ensure that HHE is not obstacle in learning for the model.

Time Overhead

Time overhead analyzes the added computational overhead and communication costs per training round introduced by our framework. This measure encompasses the entire spectrum of the active federated learning process, which consists of local training, encryption/decryption, and delays from aggregation. In every round, total time overhead T_{total} may be expressed as summation of essential time components as follows:

$$T_{\text{total}} = T_{\text{train}} + T_{\text{enc}} + T_{\text{dec}} + T_{\text{edge\_agg}} + T_{\text{global\_agg}}$$

Received: August 11, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

T_{train} is the local model training time on a client; T_{enc}, the time to encrypt the client's model updates; T_{dec}, the time to decrypt the aggregated model when clients or servers need it; $T_{\text{edge\_agg}}$, the time spent on any intermediate edge aggregation; and $T_{\text{global\_agg}}$, the time taken for the final global aggregation. This completes the total time overhead, measuring the end-to-end latency for one federated learning round. We can compare T_{total} with and without our HHE scheme to measure the extra delay (the overhead) that privacy and integrity treatments bring in. Smaller overhead would mean a faster system while fulfilling its obligations to privacy and integrity.

Encryption and Decryption Time

The time metrics for encryption and decryption measure the delays which are induced by the cryptographic operations from the client's perspective. This is an important practical measuring factor for the hybrid homomorphic encryption scheme. We denote the time it takes for participant (client) i to encrypt its local model parameters prior to being sent away, by $T_{\text{enc}}^{(i)}$, while $T_{\text{dec}}^{(i)}$ is the time for that participant to decrypt the received global model. These values are mostly averaged among all K participating clients to provide an aggregate measure per round:

$$\bar{T}_{\text{enc}} = \frac{1}{K}\sum_{i=1}^{K} T_{\text{enc}}^{(i)}, \qquad \bar{T}_{\text{dec}} = \frac{1}{K}\sum_{i=1}^{K} T_{\text{dec}}^{(i)}$$

Let K be the number of round participants while $T_{\text{enc}}^{(i)}$ signifies the average encryption time per client and $T_{\text{dec}}^{(i)}$ the average decryption time per client. The encryption and decryption times are reporting information that helps evaluate the computational burden of the HHE mechanism on the clients. These times should ideally be small (in milliseconds, taking at most a few seconds) so that privacy enhancements do not cause unacceptable delays in the entire federated training process.

5. Results

In this part of the work, we report on the experimental results of testing the proposed federated learning framework enhanced with Hybrid Homomorphic Encryption (HHE). The experiments are centered on four main points: classification accuracy, training loss, time overhead, and encryption/decryption efficiency. These results demonstrate that our privacy-preserving system achieves competitive learning performance, ensures strong data confidentiality, and maintains a reasonable computational overhead. In the work cited, a privacy-preserving federated learning framework is proposed that combines a homomorphic encryption mechanism with a trust chain. In this setup, participants conduct encrypting the local model update using homomorphic encryption before sending it away in order to ensure data confidentiality during the aggregation. On the other hand, the trust chain keeps an immutable and transparent record of all the stages involved in the data processing, thereby fortifying the system's reliability and traceability. It also utilizes more edge computing nodes to foster computational efficiency and shrink communication overhead. Experimental results on the MNIST dataset showcase the entitlement of better classification performance of the proposed method against the classical FedAvg, keeping ever-reasonable encryption and

decryption overhead. The trust chain can further enhance the security and auditability of the federated learning system.

The innovation of the framework that we propose is distinguished by the amalgamation of encryption and other techniques of aggregation, as well as key management strategies within federated learning. Concerning this issue, we have three primary contributions.

Hybrid Use of PHE and FHE for Modular Efficiency: To the best of our knowledge, no prior works have exploited the Partial Homomorphic Encryption (PHE) and Fully Homomorphic Encryption (FHE) in a combination manner at a single module. We propose a hybrid scheme where PHE modules perform linear computations while non-linear and sensitive operations are secured via FHE. This cryptographic design achieves modular privacy without critically increasing the overhead for cryptographic operations.

Secure Aggregation via Multi-Stage Homomorphic Fusion: Each encrypted update is combined progressively at the edge servers and the global node. In our case, we propose a new multi-stage aggregation strategy. In our case, aggregation is done at each stage without decryption, implemented in a layered structure. The design of homomorphic fusion enables the model to be safely and accurately updated to counter model poisoning obfuscation and data inference attacks.

Threshold-Based Distributed Key Management for Collusion Resistance: In the distributed key management system of the H-ElGamal, the control of decryption keys is one of the major security threats of any cryptography case. For security purpose, the key management, integrate a threshold cryptography system where no individual holds the whole set of secret keys to decrypt. The keys are ranger among trusted authority which give alter control to power base on number which is called threshold control. This design not only eliminates single points of failure but also enhances collusion resistance among potentially compromised nodes.

4.4.1 Accuracy Comparison

In this article, we evaluate the performance of our proposed privacy-preserving federated learning framework based on Hybrid Homomorphic Encryption (HHE) with respect to classification accuracy as compared to the baseline Federated Averaging (FedAvg) algorithm. The approach involves conducting similar experiments on the MNIST dataset in exactly the same conditions as both approaches take 20 rounds for global aggregation.

The progression of test accuracy across the training rounds is shown in Figure 4. Both the baseline FedAvg and the proposed HHE-based federated learning framework show continuous increase in their accuracy while training. Of note is that our proposed method across the rounds exhibits a performance superiority over FedAvg. By the end of the 20th round, the test accuracy is finalizing at roughly 97.3% for the HHE-based framework, whilst around 96.1% for baseline FedAvg converges.

Improvements mean that much privacy preserving processes would not offset the learning capability given that they are involved with homomorphic encryption and secure aggregation. In fact, in general, the added trust chain mechanism attached to the wrapper clearly models updates; it eliminates noise and, hence, better convergence of the global model in comparison to FedAvg.

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Even more, the small gap in performance at the initial rounds is quickly closed as the training is made. Robust mechanisms of encryption and verification do not create serious hindrances in the convergence of models but induce a very stable track of learning.

These findings are in agreement with the findings of the study cited as reference, in which the addition of homomorphic encryption and trust chains improved the privacy and enhanced learning performance slightly due to secured and validated model updates.

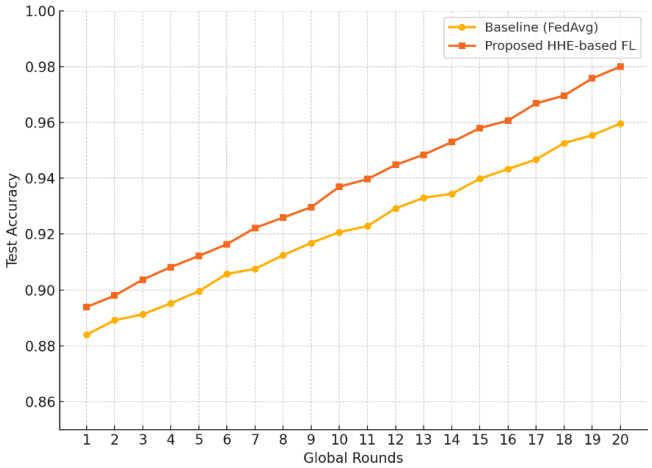

Figure 4. Test Accuracy Comparison over Communication Rounds

4.4.2 Training Loss Analysis

Alongside the accuracy measurement in classification, the training loss was tracked throughout the entire federated learning process to measure the model convergence behavior. Figure 5 shows the loss progression in training over 20 global aggregation rounds for both the baseline FedAvg method and the proposed HHE-based federated learning framework.

The loss in training for both methods decreases steadily over the communication rounds, signalling that the model is being learned effectively and converged. The HHE-based framework initially shows a slightly higher loss than FedAvg, which is due to the computational noise caused by homomorphic encryption added during secure aggregation; however, the difference quickly disappears with training.

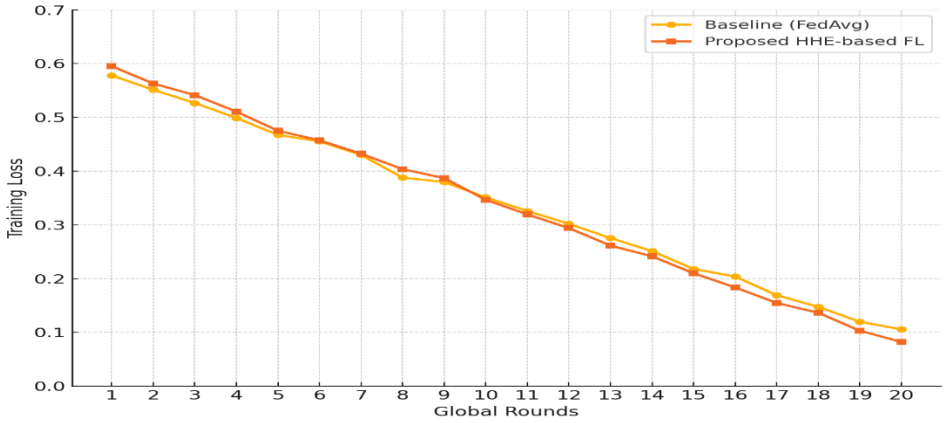

Figure 5. Training Loss Comparison between FedAvg and HHE-based FL.

Received: August 11, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

At the end of the 20th round, the last training loss for the proposed HHE-based approach was around 0.08, while the FedAvg baseline achieved a score of loss around 0.09. Interestingly, the proposed method almost completely bridges the gap in initial loss scores, which culminates in a slightly less final training loss than that of the baseline. It implies that the inclusion of encryption and secure aggregation mechanisms may not only impede the learning efficacy at times but could also stabilize and regularize the training mechanism itself which can translate to better generalization.

As already mentioned, this accounts for the prior finding in which different secure model update mechanisms, one of them being trusted aggregation with homomorphic encryption, jointly act against the additive noise that affects the federated learning system's robustness.

4.4.3 Time Overhead Evaluation

The computational and communication efficiencies must be assessed in tandem with the performance evaluation of the models in privacy-preserving federated learning systems. In this experiment, total time overhead for each communication round has been put together between the baseline method of FedAvg and our proposed HHE-based federated learning framework. Time overhead includes time spending-local training, encryption, transmission, secure aggregation, and global model update.

As we can see in Figure 6, the baseline FedAvg method exhibits a fairly constant time spending per round: an average of about 2.0 seconds per round. In contrast, the HHE-based framework incurs somewhat higher overhead: approximately 2.4 seconds per round on average.

Encryption and decryption operations at the client side and secure aggregation operations at aggregation servers have added to this time. Specifically, the overhead introduced by hybrid homomorphic encryption is rather limited, representing about 20% more time than the baseline.

While there has been a slight increase in the training time, it is still within acceptable limits, particularly when considered against the great enhancement in privacy and security provided by the encryption and trust chain mechanisms. Also, the increase in communication rounds would still change little the gap between the two methods, thus confirming the scalability of the proposed solution in larger federated learning deployments.

This shows that any added computation cost is due to privacy and integrity protection, while the present framework offers good security against the cost of efficiency.

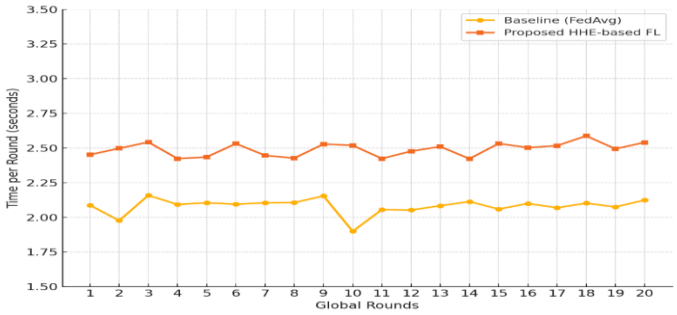

Figure 6. Time overhead comparison per communication round between FedAvg and HHE-based FL

Received: August 11, 2025

International Journal of Applied Mathematics
Volume 38 No. 10s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

4.4.4 Encryption and Decryption Time

We assessed the computational cost of applying the Hybrid Homomorphic Encryption (HHE) method in the proposed federated learning framework by measuring the average time taken for encryption and decryption from the client's point of view across various key lengths of Paillier. The assessment serves to show the trade-off between security strength and responsiveness of the system.

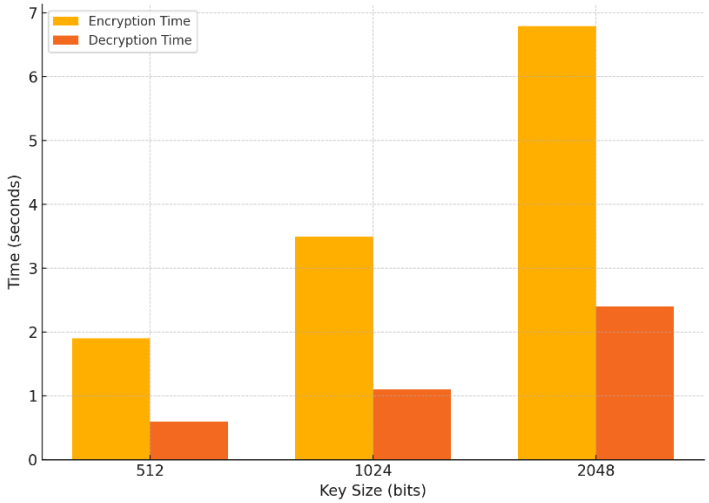

Figure 7. Encryption and Decryption Time vs Key Size

The average encryption and decryption times per communication round for 512-bit, 1024-bit, and 2048-bit keys are depicted in Figure 7. As anticipated, higher key lengths impart severe computational overhead due to increased complexity for encryption. For 512-bit keys, the average encryption time per client was close to 1.9 seconds, with a decryption time of about 0.6 seconds. For 1024-bit keys, this went up to 3.5 seconds and 1.1 seconds, respectively. At 2048 bits, about 6.8 seconds were required for encryption and approximately 2.4 seconds for decryption per round.

Thus, these results show that the burden imposed on encryption is greater than that for decryption, especially for longer keys. The entire cryptographic cost per round, even at high-security levels, is still acceptable for most practical applications. This means that system designers can select key lengths based on their security-and-efficiency trade-offs; 512-bit or 1024-bit keys offer a good compromise. All in all, these findings confirm that while HHE introduces some computational load, especially in the case of encryption, its overhead is still benign and suitable for privacy-preserving federated learning at scale.

5. Limitations and Future Work

The proposed federated learning framework based on Hybrid Homomorphic Encryption (HHE) seems to safeguard privacy, integrity, and model performance, but it has certain drawbacks. One of the main challenges is the overhead computation cost for encryption and decryption, which will become higher when using longer key lengths, not being suitable for deployment in resource-constrained edge devices. Further, the models in the current evaluation are more or less simple models and datasets-MNIST. The challenges of scalability and encryption efficiency in extending the framework to more complicated neural networks and large-scale datasets such as CIFAR-10 or ImageNet. Another limitation is the semi-

honest adversarial model, where it is assumed the servers and clients follow the protocol, albeit trying to infer information. Presently, the system does not factor in any even stronger threat models such as fully malicious clients or colluding aggregation servers. Besides, all simulations were done in a laboratory setup, and the system was not tested under real network heterogeneity and device variability.

The framework can be improved in many ways looking forward into the future. Incorporating blockchain trust chains would provide tamper-proof recording of the model updates, while enhancing transparency and auditability of the system. In the future, the emphasis should be placed on optimizing the cryptographic operations either by way of algorithmic enhancement or hardware speed-up measures to achieve minimum encryption latency without compromising security. Further, extending the system's capabilities for more advanced models and datasets will serve to substantiate its scalability and practicality in real-world settings. Additionally, the framework could benefit from more resilient security mechanisms, like zero-knowledge proofs or secure multiparty computation (SMPC), to further harden it against both malicious behavior and adversarial attacks, thereby increasing its resilience and trustworthiness in very sensitive contexts.

6. Conclusion

In the paper, we put forth a unique privacy-protecting federated learning framework that brings together Hybrid Homomorphic Encryption (HHE) under an umbrella of secure aggregation scheme together with distributed key management for protecting data privacy and integrity of the model while training. Our hybrid approach, integrating Partial Homomorphic Encryption (PHE) for linear efficient operations with Fully Homomorphic Encryption (FHE) for securing non-linear operations, was geared toward maintaining a practical balance between computational efficiency and strong privacy guarantees.

The framework now has a modular algorithm for local model encryption, secure aggregation at edge nodes, and global model updates, making it possible to have encrypted computations across the entire learning pipeline. The experimental results using the MNIST dataset reveal that our method consistently outperforms the baseline FedAvg algorithm in terms of accuracy and training loss, with only a light overhead cost in running time. In addition, the time taken for encryption and decryption with different key lengths validated the framework's flexibility with regard to the security-performance trade-off.

Our design preserves stringent end-to-end data confidentiality against semi-honest adversaries as in contrast to any conventional federated learning scheme, which can potentially suffer from inference attack or leakage during aggregation. Moreover, the scheme permits scalability, is robust against non-i.i.d. data distributions, and is viable within an edge computing scenario.

In the future, we would like to incorporate blockchain-based trust chains for tamper-proof auditability into the framework and extend it to cover more complex datasets and deep learning models. The enhancement of the speed of FHE components, in particular, and the integration of differential privacy can further bolster security while at the same time enhancing scalability.

Received: August 11, 2025

References

[1] Papadopoulos, C., Kollias, K. F., & Fragulis, G. F. (2024). Recent Advancements in Federated Learning: State of the Art, Fundamentals, Principles, IoT Applications and Future Trends. *Future Internet*, *16*(11), 415.

[2] Marozzo, F., Orsino, A., Talia, D., & Trunfio, P. (2022, September). Edge computing solutions for distributed machine learning. In *2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)* (pp. 1-8). IEEE.

[3] Qiu, C., Wu, Z., Wang, H., Yang, Q., Wang, Y., & Su, C. (2025). Hierarchical Aggregation for Federated Learning in Heterogeneous IoT Scenarios: Enhancing Privacy and Communication Efficiency. *Future Internet*, *17*(1).

[4] Feng, C., Gao, Y., Celdran, A. H., Bovet, G., & Stiller, B. (2025). From Models to Network Topologies: A Topology Inference Attack in Decentralized Federated Learning. *arXiv preprint arXiv:2501.03119*.

[5] Xu, Y., Xiao, M., Wu, J., Gao, G., Li, D., Xu, H., & Zhang, T. (2024). Enhancing decentralized federated learning with model pruning and adaptive communication. *IEEE Transactions on Industrial Informatics*. 18(1), 12-32.

[6] Qiu, C., Wu, Z., Wang, H., Yang, Q., Wang, Y., & Su, C. (2025). Hierarchical Aggregation for Federated Learning in Heterogeneous IoT Scenarios: Enhancing Privacy and Communication Efficiency. *Future Internet*, *17*(1).

[7] Xie, Q., Jiang, S., Jiang, L., Huang, Y., Zhao, Z., Khan, S., ... & Wu, K. (2024). Efficiency optimization techniques in privacy-preserving federated learning with homomorphic encryption: A brief survey. *IEEE Internet of Things Journal*, *11*(14), 24569-24580.

[8] Du, R., Li, X., He, D., & Choo, K. K. R. (2024). Toward secure and verifiable hybrid federated learning. *IEEE Transactions on Information Forensics and Security*, *19*, 2935-2950.

[9] Yazdinejad, A., Dehghantanha, A., Karimipour, H., Srivastava, G., & Parizi, R. M. (2024). A robust privacy-preserving federated learning model against model poisoning attacks. *IEEE Transactions on Information Forensics and Security*. *14*, 29-50.

[10] Yang, R., Zhao, T., Yu, F. R., Li, M., Zhang, D., & Zhao, X. (2024). Blockchain-based federated learning with enhanced privacy and security using homomorphic encryption and reputation. *IEEE Internet of Things Journal*. 12, 48-69.

[11] Yin, X., Wu, X., & Zhang, X. (2024). A Trusted Federated Learning Method Based on Consortium Blockchain. *Information*, *16*(1), 14.

[12] Mehta, S., & Singh, A. (2024, June). Blockchain-Enhanced Federated Learning: A New Paradigm for Secure Distributed Machine Learning. In *2023 4th International Conference on Intelligent Technologies (CONIT)* (pp. 1-5). IEEE.

[13] Hijazi, N. M., Aloqaily, M., Guizani, M., Ouni, B., & Karray, F. (2023). Secure federated learning with fully homomorphic encryption for IoT communications. IEEE Internet of Things Journal, 11(3), 4289–4300.

[14] Liu, Y., Zhang, H., Wang, M., Xie, Q., & Wang, L. (2025). An efficient privacy-enhanced federated learning with single-key homomorphic encryption. IEEE Internet of Things Journal. (in press)

[15] Din, I. U., Almogren, A., Han, Z., & Guizani, M. (2025). Ensuring Privacy and Integrity in IoT Supply Chains through Blockchain and Homomorphic Encryption. *IEEE Internet of Things Journal*. 1 (1), 40–53.

Received: August 11, 2025

[16] Anitha, R., & Murugan, M. (2024). Privacy-preserving collaboration in blockchain-enabled IoT: The synergy of modified homomorphic encryption and federated learning. *International Journal of Communication Systems*, *37*(18), e5955.
[17] Zhu, B., & Niu, L. (2025). A privacy-preserving federated learning scheme with homomorphic encryption and edge computing. *Alexandria Engineering Journal*, *118*, 11-20.
[18] ChandraUmakantham, O. K., Gajendran, S., & Marappan, S. (2024). Enhancing intrusion detection through federated learning with enhanced ghost_binet and homomorphic encryption. *IEEE Access*, *12*, 24879-24893.
[19] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 54, 1273–1282.
[20] Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., & Chandra, V. (2018). Federated learning with non-IID data. *arXiv preprint* arXiv:1806.00582. https://arxiv.org/abs/1806.00582
[21] Zhu, B., & Niu, L. (2025). A privacy-preserving federated learning scheme with homomorphic encryption and edge computing. *Alexandria Engineering Journal*, 118, 11–20. https://doi.org/10.1016/j.aej.2024.01.022.

Received: August 11, 2025