*Journal Name*

Research Paper

# Privacy-Preserving Deep Federated Learning on the Edge Using Homomorphic Encryption and Secure Multiparty Computation

**Noman Aasif Gudur[1], Mohamed El-Dosuky[2,3], Sherif Kamel[2,4]**

[1] *Next Tech Lab, SRM University - AP, Andhra Pradesh, India*
[2] *Department of Computer Science, Arab East Colleges, Riyadh, Saudi Arabia*
[3] *Department of Computer Science, Faculty of Computers and Information, Mansoura University, Egypt*
[4] *Department of Communications and Computer Engineering, October University for Modern Sciences and Arts, Egypt*

*Article history*
Received: 21 January 2025
Revised: 7 April 2025
Accepted: 8 June 2025

*Corresponding Author:
Mohamed A. El-Dosuky,
Dept. of Computer Science,
Arab East Colleges,
Riyadh, Saudi Arabia
Email:
maldosuky@arabeast.edu.sa

**Abstract:** The increasing volume of consumer data necessitates reliable edge devices for personalized user experiences. Federated Learning (FL) offers a state-of-the-art approach to decentralized machine learning by leveraging data distributed across multiple client devices. However, user data privacy remains vulnerable to corruption through feature heterogeneity and malicious attacks. While several privacy-preserving techniques have been previously implemented, they suffer from implementation constraints and limited robustness against sophisticated attacks. This paper proposes a deep convolutional neural network mechanism that enhances privacy preservation in FL by combining Homomorphic Encryption (HME) and Secure Multiparty Computation (SMC). The proposed approach is validated through model verification on the CIFAR-100 dataset and a healthcare diabetes dataset case study. Results demonstrate that the proposed mechanism outperforms existing privacy protection methods, particularly against backdoor attacks. By ensuring stronger privacy guarantees, this approach facilitates broader adoption of FL technology across privacy-sensitive domains.

**Keywords:** Federated Learning, Homomorphic Encryption, Secure Multiparty Computation, Privacy-Preserving Machine Learning, Backdoor Attacks, Edge Computing, Deep Learning

## Introduction

Federated Learning (FL) enables model training across multiple decentralized devices while preserving data locality, thereby improving privacy protection, reducing data transfer costs, and minimizing computational overhead (Zhang et al., 2021). In FL, each participating device develops a local model using its private data, and these local models are subsequently aggregated to construct a global model. Despite these architectural advantages, FL remains vulnerable to various security threats that can compromise user data privacy and model integrity (Bouacida & Mohapatra, 2021).

Backdoor attacks represent a particularly insidious threat, wherein adversaries inject malicious patterns during local training that cause the global model to produce incorrect predictions or decisions when specific triggers are present (Lyu et al., 2020). In visual datasets such as CIFAR-100, backdoor triggers can be subtle, such as small white dots or texture alterations, making detection challenging while maintaining attack effectiveness. The stealth mechanisms employed in model poisoning attacks across multiple clients remain an open research problem (Zhou et al., 2021), necessitating more robust defense strategies.

To defend against backdoor attacks, robust aggregation techniques, data augmentation techniques, and a mixture of clean and adversarial examples can be used. HME and SMC are used to resolve these attacks. Homomorphic Encryption (HME) and Secure Multi-Party

Computation (SMC) are essential in federated learning (FL) because existing privacy-preserving methods, such as differential privacy and trusted execution environments (TEEs), are insufficient to fully protect against advanced attacks (Xie et al., 2024). Differential privacy, while effective in limiting information leakage, introduces noise that can significantly reduce model accuracy (El Ouadrhiri & Abdelhadi, 2022). TEEs rely on hardware security, which can be vulnerable to side-channel attacks and may not scale efficiently across distributed systems (Cerdeira et al., 2020). In contrast, HME enables computations on encrypted data without requiring decryption, ensuring that sensitive information remains confidential even during model aggregation. Similarly, SMC allows multiple parties to collaboratively compute a function without revealing their private inputs, offering strong security guarantees even in the presence of semi-honest or colluding adversaries. These cryptographic techniques provide robust and scalable privacy protection, making them indispensable for securing FL against threats like model inversion, gradient leakage, and membership inference attacks.

The problem can be stated mathematically as follows: let there is a dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$ and there is a model $f_\theta$ that is parametrized by θ to be trained. There are m parties, each holding a portion $D_j$ of the dataset.

To model SMC secret sharing, each data value x is shared among m parties: $x = x_1 + x_2 + \cdots + x_m$ . Each $x_j$ is a random share such that no individual $x_j$ reveals information about x.

To model HME computation on encrypted data, assume an additive homomorphic encryption scheme Enc (), such that:

$$\text{Enc}(x_1) + \text{Enc}(x_2) = \text{Enc}(x_1+x_2) \quad (1)$$

If x is the input and w is the model weight, then encrypted linear model output is:

$$\text{Enc}(w^\top x) = w^\top \text{Enc}(x). \quad (2)$$

Since $x_j$ alone is statistically independent of x, and $\text{Enc}(x_j)$ reveals nothing due to encryption, then:

$$I\left(x; \{x_j\}_{j=1}^{m}\right) = 0 \quad (3)$$

and

$$I\left(x; \{Enc(x_j)\}\right) = 0 \quad (4)$$

So mutual information is zero, i.e., privacy is preserved.

Let model update rule be:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta \mathcal{L}(f_\theta(x), y) \quad (5)$$

where $\eta$ denotes the learning rate, $\mathcal{L}$ is the loss, and $\nabla\mathcal{L}$ is computed on secret shares via SMC.

Since

$$\nabla\mathcal{L}(Enc(x)) = Enc(\nabla\mathcal{L}(x)) \quad (6)$$

Aggregation over shares yields the correct gradient:

$$\nabla\mathcal{L}(x) = \sum_{j=1}^{m} \nabla\mathcal{L}(x_j) \quad (7)$$

The update becomes:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \sum_{j=1}^{m} Dec(\nabla\mathcal{L}(x_j)) \quad (8)$$

Since decryption and summation are exact, accuracy is preserved.

Key contributions of this paper can be summarized as follows: First, the paper highlights the various security challenges in implementing FL models on edge devices and proposes solutions to address them. Second, in order to improve privacy-preserving in FL, this paper proposes a deep convolutional neural network technique that combines secure multiparty computation (SMC) (Gafni et al., 2024) with Homomorphic Encryption (HME) (Acar et al., 2018). Third, model verification is performed using the CIFAR-100 dataset. The results demonstrated that by considering backdoor attack, the proposed mechanism performs better than other techniques for privacy protection.

The next section overviews the understanding needed to apprehend the new realms of privacy-preserving FL in the edge. Subsequent sections provide a literature survey before proposing the methodology and implementation. Results and conclusion are provided at the end.

## Preliminaries

*Privacy in Edge computing*

Edge computing addresses data privacy and time requirements (Aslanpour et al., 2021), offering high bandwidth and low latency (Mitra et al., 2020). It enhances scalability due to smaller devices. A learning environment on collaboration around network margins can forecast failures for safe group driving (Lu, Yao, & Shi, 2019). Smart hubs protect privacy in household settings by regulating access to sensitive data (Zavalyshyn, Duarte, & Santos, 2018). Autonomous decentralized learning systems use FL strategies for intrusion detection (Nguyen et al., 2019). Edge computing infrastructure is used in automotive applications for improved engagement (Xie, Koyejo, & Gupta, 2019). Task offloading scheduling techniques address privacy concerns (Zhao et al., 2018).

*Secure Multiparty Computation (SMC)*

SMC is a concept where multiple parties work together to calculate a common function without disclosing their private inputs (Alon, Omri, & Paskin-Cherniavsky, 2020). The approach involves determining the method of update aggregation and the aggregation frequency.

As shown in Fig. 1, SMC works by updating models at edge devices, encrypting them, and securely aggregated. The central server updates the centralized model, sending new updates iteratively until the desired accuracy is achieved. This Figure illustrates the process of FL, a decentralized approach to training machine learning models while preserving data privacy. In this setup, multiple client devices (such as smartphones) locally train models using their private data. Rather than sending raw data to a central server, each client computes model updates or gradients, which are then encrypted to ensure privacy. These encrypted updates are sent to a secure aggregation mechanism, which combines them without revealing individual client information. The aggregated result is used by the central server to update the global model. Finally, the updated global model is sent back to the clients, and the process repeats. This architecture enhances data security and privacy while enabling collaborative model training across distributed devices.

SMC properties include correctness, input independence, privacy, output delivery assurance, and fairness (Canetti, 2000). They preserve that each party receives accurate output, ensuring no one can thwart the outcome. In an auction, the highest bidder always wins, and no one can thwart the outcome. Privacy ensures no party gains knowledge beyond the prescribed output. Output delivery assurance ensures honest parties can collect their output without interference from corrupted parties.

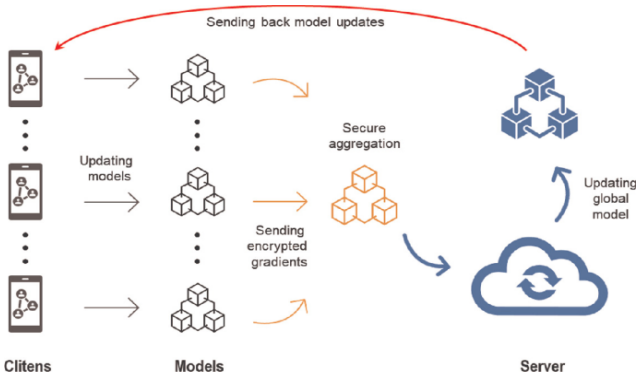


**Fig. 1.** Secure Multiparty Computation

*Homomorphic Encryption (HME)*

HME is an encryption technique introduced in 2009 that allows data to be processed in an encrypted state without requiring a secret key (Alloghani et al., 2019). This method uses mathematical functions to encrypt data without decryption, ensuring privacy and security. The data remains private even when processed by multiple parties. HME has been transformed by cloud computing and storage, but analyzing encrypted data requires a secret key, which poses security concerns. Additionally, data owners may need to work locally, which can be logistically challenging and costly.

Homomorphic encryption, including PHE, SHE, and FHE that stands for partially, somewhat, and fully homomorphic encryption (Acar et al., 2018). PHE technique allows secure computations on encrypted data without decryption, and privacy-preserving (Ma, Naas, Sigg, & Lyu, 2022).

PHE is particularly useful in federated learning, where a central model is trained using dispersed information from multiple clients.

The Paillier cryptosystem is a PHE technique. This encryption scheme is asymmetric and relies on the assumption that composite residuosity is difficult to determine. It is additively homomorphic, meaning that it supports the addition of encrypted messages and decryption of the sum.

In Fig. 2, the Paillier cryptosystem is implemented in (Shah, Zhang, Hu, & Yu, 2019). The CIFAR-100 dataset involves pre-processing images, encrypting them using the Paillier cryptosystem, and transmitting them to a global server. The central server produces encrypted gradients, which are decrypted and transmitted back to edge devices. The images are then tested on local data. The global server then performs homomorphic multiplication and addition operations on the encrypted data.

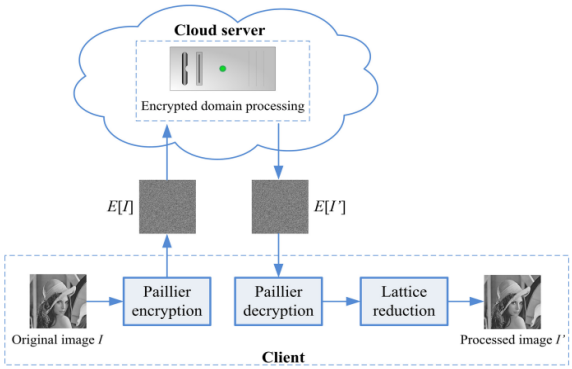


**Fig. 2.** Paillier Cryptosystem

*Model Verification*

Model verification is a process that tests a trained model on unused data to ensure it performs as expected and is not compromised (Zhang, Lu, Qiu, Gui, & Shi,

2023). In FL, decentralized data is used, presenting challenges. Techniques like differential privacy, which introduces random noise to prevent user information disclosure, can help evaluate model accuracy and prevent overfitting (Sun, Wang, Shu, Liu, & Zhao, 2019). Model verification is crucial in detecting potential backdoor attacks and other model issues in FL (Shokri & Shmatikov, 2015).

## Related Work

FL balances global parameter aggregation and local updates, minimizing loss function and synchronizing training samples and features (Lim et al., 2020). Table 1 summarizes recent poisoning attacks, which are dichotomized into model and data poisoning. In model poisoning, there could be a scenario satisfying both stealth and persistence (Zhou et al., 2021), another scenario not satisfying neither of them (Bagdasaryan, Veit, Hua, Estrin, & Shmatikov, 2020), and a scenario satisfying only stealth (Bhagoji, Chakraborty, Mittal, & Calo, 2019).

Backdoor attacks can introduce bias into models, leading to incorrect predictions or decisions (Lyu et al., 2020). They are particularly effective in visual datasets like CIFAR-100. To defend against these attacks, robust aggregation techniques, data augmentation, and HME and SMC are employed. A recent paper gives a comprehensive analysis of the most recent backdoor attacks and countermeasures in FL (Gong, Chen, Wang, & Kong, 2022).

Differential Privacy (DP), Secure Aggregation (SecAgg), and Trusted Execution Environments (TEEs) are complementary technologies used to enhance privacy and security in data-driven systems. DP introduces mathematical noise to data or model updates, ensuring that the output of a computation does not compromise the privacy of individual data points, even in the presence of adversaries (Zhu et al., 2020). SecAgg enables multiple parties to collaboratively train models by securely aggregating their updates without exposing individual contributions, using cryptographic techniques to ensure confidentiality throughout the process (Fereidooni et al., 2021). TEEs, such as Intel SGX, provide isolated hardware environments where sensitive computations can be executed securely, protecting against attacks even from privileged system software (Chakrabarti et al., 2020).

These technologies play a crucial role in privacy-preserving machine learning and federated learning frameworks. They are compared in Table 2.

**Table 1:** Recent poisoning attacks

| Attack Category | Scenario | Stealth | Persistence | Reference |
|---|---|---|---|---|
| **Model Poisoning** | FL backdoor | No | No | (Bagdasaryan, Veit, Hua, Estrin, & Shmatikov, 2020) |
| | Adversarial FL | Yes | No | (Bhagoji, Chakraborty, Mittal, & Calo, 2019) |
| | Deep attack on FL | Yes | Yes | (Zhou et al., 2021) |
| **Data Poisoning** | Recommendation system | No | No | (Fang, Gong, & Liu, 2020) |
| | Clean-label attack | Yes | No | (Shafahi et al., 2018) |
| | Regression learning | No | No | (Jagielski et al., 2018) |
| | Deep learning | No | No | (Muñoz-González et al., 2017) |

**Table 2:** Other privacy-preserving techniques

| Technique | Accuracy | Privacy Level | Computational Cost |
|---|---|---|---|
| **DP** | Moderate to High (depends on noise level) | Strong theoretical guarantees | Medium to High (depends on noise and mechanism) |
| **SecAgg** | High (aggregates encrypted data without decryption) | Strong (data remains encrypted during aggregation) | High (cryptographic operations) |
| **TEEs** | High (processes raw data inside secure enclave) | Strong (hardware-based isolation) | High (performance overhead due to enclave operations) |

## Proposed Method

In this paper, an implementation for combining both SMC and HME is done. Also, model verification is applied to improve the model performance keeping huge data scenarios in the picture. Paillier encryption is used on edge devices in privacy-preserving systems due to its efficiency in additive operations, making it well-suited for aggregating encrypted updates in Federated Learning while maintaining privacy. It's lightweight, which is ideal for resource-constrained devices. BGV encryption, on the other hand, is employed on the central server because it supports both addition and multiplication of encrypted data, enabling more complex computations like model training and aggregation of gradients. While BGV is more computationally demanding, it is suitable for the server's higher resource capacity.

### *Dataset*

The CIFAR-100 dataset contains color images that are 60,000 in number, with each image being 32x32 pixels and belonging to one of 100 classes. There are 20 superclasses, each consisting of 5 classes, and 600 images per class. All images in the dataset contain two labels, one that defines the superclass of the image and the other defining the class of the label, the former is called the coarse label, and the latter fine label. Each class is again classified as 500 training images and 100 testing images. For instance, one superclass is flowers which have orchids, roses, sunflowers, etc as classes in it. Sample of CIFAR-100 dataset can be seen in Fig. 3.

CIFAR-100 was chosen over alternative datasets like MNIST or ImageNet because it offers a balanced level of complexity that is suitable for evaluating deep learning models without overwhelming computational resources. Unlike MNIST, which contains simple grayscale images of handwritten digits, CIFAR-100 consists of 32×32 color images across 100 diverse classes, making it a more challenging and realistic benchmark for image classification tasks. At the same time, CIFAR-100 is significantly smaller and more manageable than ImageNet, which involves high-resolution images and requires extensive computational power and storage. This makes CIFAR-100 an ideal middle ground,complex enough to test model performance and generalization, yet lightweight enough to allow rapid experimentation, especially in privacy-preserving FL setups where encryption and decryption add computational overhead.

### *Convolutional Neural Networks (CNNs)*

CNNs are deep learning neural networks used for images and video data. Drawing inspiration from the visual cortex, CNNs extract abstract features, creating complex feature hierarchies through feature extraction and classification.

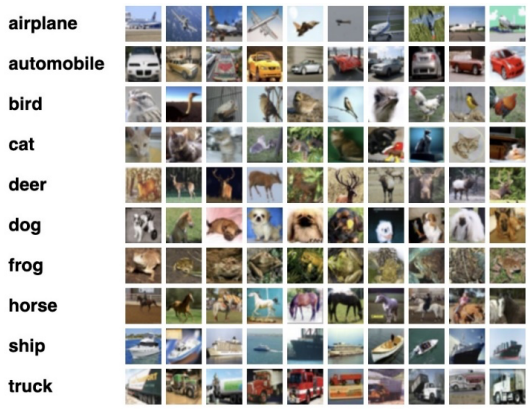


**Fig. 3.** CIFAR-100 dataset

CNNs were chosen over more complex architectures like Transformers or ResNets due to their efficiency, simplicity, and suitability for small image datasets like CIFAR-100. CNNs are specifically designed to capture local spatial features in images through convolutional filters, making them highly effective for image classification tasks. Compared to deeper architectures, CNNs require fewer parameters and less computational power, which is particularly beneficial when working with limited data and resource-intensive processes like homomorphic encryption (HE). Additionally, CNNs are easier to implement, debug, and adapt for privacy-preserving learning, as their operations are more compatible with the mathematical constraints of encrypted computation. Given these advantages, CNNs provide a balanced trade-off between performance and practicality in this setup.

The pictorial representation of the proposed CNN architecture can be seen as shown in Figure 4, that illustrates a CNN architecture designed for image classification tasks involving input images of size 32×32×3, typical of the CIFAR-10 dataset. The network begins with three convolutional blocks, each consisting of two Conv2D layers followed by ReLU activation functions, a MaxPooling layer for spatial down-sampling, and a Dropout layer to prevent overfitting. These blocks progressively extract hierarchical features from the input image. After the convolutional stages, the output is flattened into a one-dimensional vector and passed through a dense (fully connected) layer with ReLU activation, followed by another Dropout layer. Finally, the network ends with a dense layer using a Softmax activation function to produce class probabilities for classification. This architecture balances depth and regularization, making it suitable for robust image recognition.

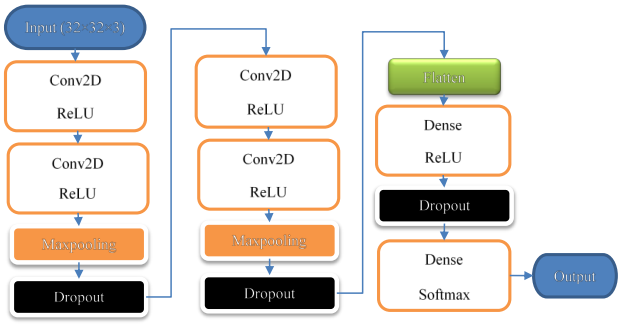


**Fig. 4.** Proposed CNN Architecture

## Implementation

### *Data Pre-Processing*

Data pre-processing is a crucial step in the training process, preparing the data for learning. It involves normalizing each pixel value by 255, enhancing data diversity, and standardizing the data to prevent overfitting. Data augmentation and standardization are also used to increase diversity and ensure stability. The dataset is divided into multiple parties, ensuring decentralization and preventing central entities from accessing all samples, which is essential for federated learning.

### *Model Initialization*

The encryption of each party of the data is done by using the SMC. The encryption is done by using a PHE scheme called Paillier cryptosystem so that the data can be securely transmitted to the central server without revealing its content. The central server then initializes the CNN model with the same architecture of the edge CNN architecture discussed above. However, this model is encrypted using a different PHE scheme called BGV cryptosystem.

Data pre-processing is a crucial step in the training process of a model. It involves normalizing the data by dividing each pixel value by 255, enhancing its diversity, and standardizing the pixel values to ensure stability. The dataset is divided into multiple parties, each with equal samples of the training data, to maintain decentralization and prevent central access. After training, the central server sends updated model parameters to clients using the same SMC protocol. Each party decrypts the encrypted model parameters using the private key for addition and uses the decrypted parameters for local inference. The Brakerski-Gentry-Vaikuntanathan (BGV) cryptosystem is used for encryption, which performs multiplication and summation operations on encrypted data. The central server performs homomorphic encryption on the encrypted model updates, secure aggregation, and decryption. The final model is encrypted using the public key for addition, and the final model is transmitted to clients for evaluation.

### *Secure Model Aggregation*

The global server receives the data that is being encrypted by utilizing the secure multiparty computation protocol. The central server performs homomorphic encryption on the encrypted data using the Paillier cryptosystem and then performs secure aggregation of the encrypted data using the SMC protocol. This allows the central server to compute the average of the gradients of each party's data without ever seeing the data itself.

### *Secure Model Update*

The global server transfers the model being encrypted to each party using the SMC protocol. Each party decrypts the model using its secret key and performs local model updates on its encrypted data. The party then encrypts the updated model using the PHE scheme used in model initialization and sends it back to the server.

### *Proposed Algorithm*

The proposed algorithm is detailed in Fig. 5. It works in three stages as follows:

**Stage 1: Secure multiparty computation**

The whole implementation is done in three stages. In stage 1, the necessary packages are imported and the CIFAR-100 dataset is acquired and pre-processed accordingly. Secure multiparty computation is implemented by partitioning the whole dataset into three parties and then encryption by using the Paillier cryptosystem.

**Stage 2: Homomorphic Encryption (HME)**

In stage 2 of the implementation, homomorphic encryption is implemented by transmitting the encrypted data between parties. Then the edge CNN model is implemented because CIFAR-100 is a dataset of images. At this level, the central model takes the encrypted gradients as images, and then the BGV protocol is initialized in the central model. The outcome of the central model is encrypted which is then decrypted. This is an encrypted version because the input data for the model is also an encrypted version. These decrypted gradients from the central model are provided to edge devices where the Paillier encrypted which is decrypted for further results.

### Stage 3: Model Verification and Testing

In stage 3, model verification is done after decrypting the trained model using the decryption of the BGV cryptosystem. The validity of the decrypted model is then evaluated by leveraging the data that was not utilized for the purpose of training. At last, model evaluation on the test dataset is done.

The algorithm involves edge devices training a local model on a training dataset, which is decrypted using a homomorphic encryption scheme. The global server then aggregates the encrypted models, divides them by the number of edge devices, and sends the average weight back to each edge device. The local model is updated, encrypted, and sent back to the global server. If the model passes a verification test, it is sent to each edge device. If the model fails, the process is repeated. The algorithm uses a pseudocode to import necessary packages, load the dataset CIFAR-100, secure multiparty computation, implement an edge CNN model, and evaluate the model on a separate dataset for validation.

The proposed method combines CNN, FL, HME, and SMC on the CIFAR-100 dataset, key generation involves each participating client generating a pair of public and private keys for HE. The public keys are shared among all participants, while the private keys remain local. Parameter selection includes choosing appropriate encryption schemes such as Paillier or BFV for HE, considering the trade-off between encryption efficiency and security. In the FL setup, clients compute local gradients or model updates, which are encrypted using HME before being aggregated by the central server. The aggregation occurs securely under SMC protocols, ensuring that no party learns about the data of others. After the model updates are aggregated, clients use their private keys to decrypt the aggregated gradients or model parameters. Decryption is performed using the private keys to transform the encrypted model updates back to the original domain, allowing clients to update their local models while preserving the privacy of their data.

## Results & Discussion

The SMC mechanism gives relatively greater accuracy when compared to other mechanisms that are implemented. The computations in SMC are relatively less complex than HME and the combination variant. The HME mechanism gives better accuracy and provides a good amount of privacy for edge devices. The combination of SMC and HME is the best model in terms of privacy. This is because this combination can resist backdoor attacks. The accuracy of this model is relatively less because of the HME which is computationally costly.

The combination involves several calculations which is the reason behind the reduction of accuracy and promotion of privacy. One point to remember is the trade-off between the accuracy of the model and the privacy concern. While promoting privacy, accuracy should not be neglected which is one of the reasons behind us choosing the HME mechanism. Both accuracy and privacy of the model are to be maintained in such a way that the optimality of FL is served for the users.

In the context of combining deep FL, HME, and SMC, several potential threat models should be considered. The honest-but-curious adversary follows the protocol but attempts to infer private information from encrypted data or computations. HME and SMC protect against this threat by ensuring data remains secure during processing, preventing unauthorized access. However, the malicious server could manipulate model updates or introduce backdoors, undermining the integrity of the global model. While HME protects raw data, SMC may not fully prevent such attacks, especially during the aggregation phase, requiring additional defence mechanisms. Finally, colluding clients could intentionally submit malicious updates, such as introducing backdoors, to compromise the model. Although HME and SMC secure data privacy, they do not prevent the behavior of colluding clients, necessitating robust aggregation techniques or anomaly detection to defend against this type of threat.

Backdoor attack is implemented by introducing a malicious client into the FL process. This client injects a trigger pattern (such as a specific pixel patch) into a subset of its local training data and intentionally mislabels these samples to associate the trigger with a target class. The CNN model on the malicious client learns this association during local training. To evade detection, the client encrypts its poisoned model updates using HE or splits them using SMC, ensuring the central server cannot inspect or analyze individual contributions. These concealed, poisoned updates are then aggregated with honest client updates to form the global model. As a result, the final model behaves normally on clean data but misclassifies inputs containing the trigger pattern, successfully embedding the backdoor while remaining hidden due to the privacy-preserving mechanisms in place.

As the authors implemented the combination of SMC with partial homomorphic encryption to get rid of backdoors and verified by incorporating images via a backdoor in the dataset. After the model is trained, the authors incorporated various types of images to test the effectiveness of the privacy-preserving approach. This was done by adding a few images to the previously partitioned test data before testing. It is observed that the model is more resistant to backdoors than the other two mechanisms implemented.

**DOI: 10.3844/jcssp.2025.2581.2592**

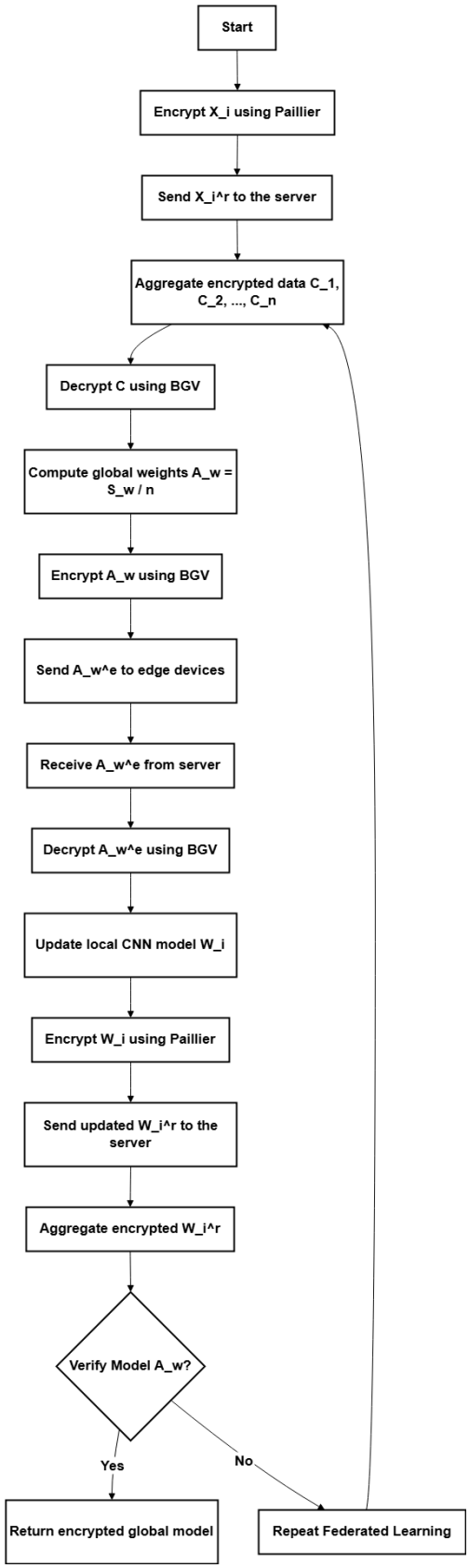


**Fig. 5.** Detailed Flowchart

**Table 3:** Different Mechanisms and their accuracies using CNN

| No | Approach to the CNN model (before applying backdoor) | | |
|---|---|---|---|
| | ***Mechanisms*** | ***Accuracy*** | ***Privacy*** |
| 1 | SMC | 84.3% | Low |
| 2 | HME | 81.9% | Very Low |
| 3 | SMC + HME | 76.7% | Very High |

**Table 4:** Different Mechanisms and their accuracies, after using Backdoor

| No | Applying backdoor on CNN model | | |
|---|---|---|---|
| | ***Mechanisms*** | ***Accuracy*** | ***Privacy*** |
| 1 | SMC | 72.3% | Low |
| 2 | HME | 68.4% | Very Low |
| 3 | SMC + HME | 70.3% | High |

When the backdoor has verified, the CNN gave exceptional results as the combination of SMC and HME is highly resistant to the backdoors. In Table 4, there is a huge difference in the accuracies of SMC and HME when compared to Table 3 above, this is because alone SMC and HME are slightly prone to backdoor attacks which is the drawback. The combination of SMC and HME is highly resistant to the backdoor attack as the accuracy reduction is much less from Table 3 to Table 4.

Fig. 6 provides runtime analysis of encryption and decryption steps on CIFAR-100.

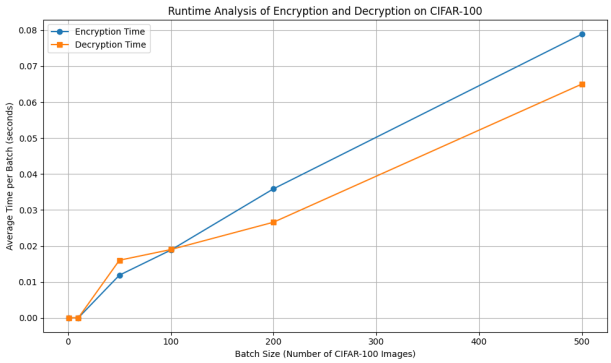


**Fig. 6.** Runtime Analysis of Encryption and Decryption steps

In real-time FL applications, such as those in IoT and healthcare, computational overhead introduced by secure techniques like secure multiparty computation and homomorphic encryption can significantly impact system performance. These privacy-preserving methods, while essential for protecting sensitive data during model training, often require extensive computation and memory resources due to complex cryptographic operations. This leads to increased latency in model updates, higher energy consumption, critical for battery-powered IoT devices,

and reduced responsiveness in time-sensitive healthcare scenarios. Consequently, the trade-off between security and efficiency becomes a critical challenge, necessitating optimized implementations or hardware acceleration to maintain real-time processing capabilities without compromising data privacy.

Fig. 7 provides a comparison among privacy techniques with excessive noise on CIFAR-100, for the first 20 epochs. It shows that the proposed method outperforms those approaches.

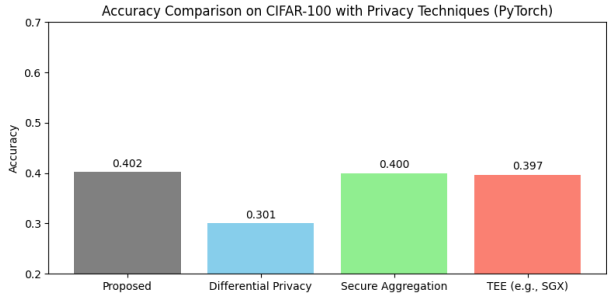


**Fig. 7.** Comparison among Privacy Techniques with noise on CIFAR-100 for the first 20 epochs

## Case Study

Fig. 8 illustrates a secure cloud-based healthcare system that utilizes public-key cryptography to protect patient data. Initially, the hospital collects plaintext data from the patient, such as symptoms or medical history. This data is then encrypted using a public key to ensure confidentiality before being transmitted to the cloud. In the cloud, encrypted data is processed, often through machine learning models or predictive analytics, and the results are also encrypted before being returned to the hospital. Only authorized individuals, such as the doctor or patient, can decrypt the returned results using a private key. This end-to-end encryption process ensures that sensitive patient information remains secure and private throughout data transmission, storage, and processing in the cloud.

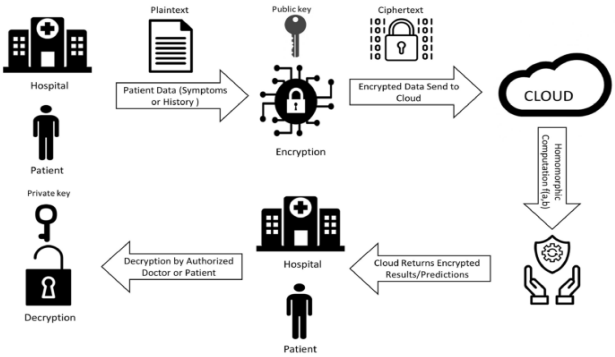


**Fig. 8.** Case Study in Healthcare

Fig. 9 illustrates the accuracy of applying the proposed methodology on Diabetes Dataset, downladable from the URL: https://archive.ics.uci.edu/ml/machine-learning-databases/00529/diabetes_data_upload.csv. It is observed that the accuracy is high from early epochs.

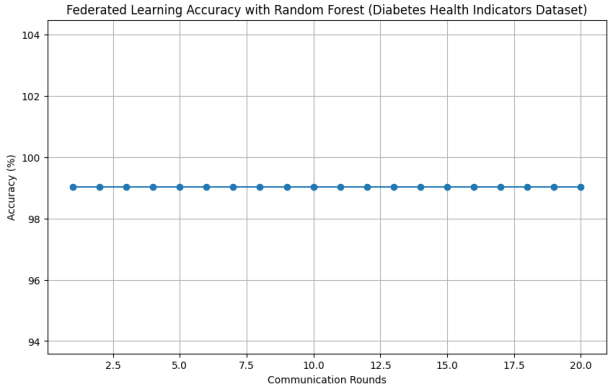


**Fig. 9.** Accuracy of Diabetes Dataset

In Membership Inference Attack Resistance, the confidence score of the model for each prediction is calculated and plot the ROC curve to assess the resistance to membership inference. Membership Inference Attacks are a type of privacy attack where an adversary aims to determine whether a particular data point was part of the model's training set. These attacks exploit the difference in the model's behavior when predicting on training data versus out-of-distribution or test data. The attack focuses on the confidence score output by the model, which can give insights into whether a sample is from the training set or not. Higher AUC indicates that the model is more vulnerable to membership inference (i.e., an attacker can better distinguish if a sample was part of the training data). Lower AUC means the model is more resistant. Fig. 10 shows the Membership Inference Attack Resistance on the Diabetes dataset. With lower value of the AUC, this indicates that the proposed model is more resistant.

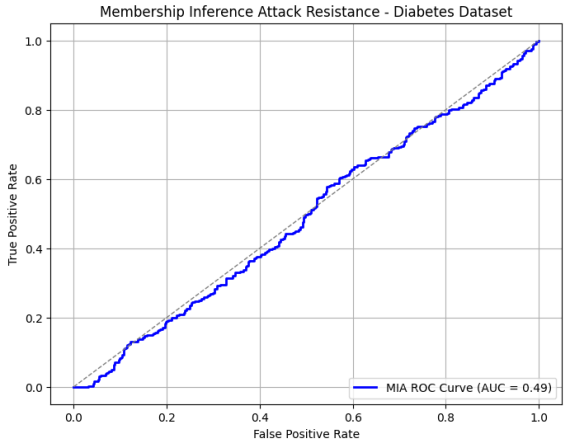


**Fig. 10.** Membership Inference Attack Resistance

Deploying HME and SMC in real-world FL systems presents several critical challenges. First, both techniques impose significant computational and communication overheads. HME, while allowing computations on encrypted data, is substantially slower than operations on plaintext and increases data size, which strains memory and bandwidth. Similarly, SMC protocols require multiple rounds of interaction between clients, further elevating latency and communication costs. These overheads limit scalability, particularly when applied to large-scale models or numerous clients. Moreover, the limited support for complex operations (e.g., non-linear activations in deep learning) makes integration into modern ML frameworks difficult. The trade-off between privacy and model performance is another obstacle, as tighter privacy often reduces accuracy or prolongs training. In addition, implementing HME and SMC typically demands specialized cryptographic expertise, custom libraries, and significant engineering effort. These challenges are compounded in resource-constrained settings such as mobile or IoT environments, where energy consumption becomes a bottleneck. Finally, regulatory uncertainties, lack of standardization, and the assumptions about trust and security in participating parties further complicate deployment. Together, these factors make the practical use of HME and SMC in FL a complex but crucial frontier for privacy-preserving AI.

## Conclusion & Future Work

This paper proposed a deep convolutional neural network mechanism to enhance privacy-preserving in FL by combining the use of HME and SMC. In this paper, different kinds of encryption techniques like Paillier cryptosystem, and BGV cryptosystems are used for the edge devices and central server separately. Usage of other available techniques is encouraged since both of the above-used techniques are computationally costly. Also, the usage of partial homomorphic encryption is made because of its efficiency. However, the fully homomorphic encryption scheme can also be tried for efficient results, only when resources are available. Different CNN architecture with even more complexity might make the model even more efficient.

## Acknowledgment

We extend our sincere thanks to the administration of both SRM University and Arab East Colleges; without their support, this research would not have been possible.

## Funding Information

This research did not receive any funding.

## Author's Contributions

**Noman Aasif Gudur**: Development, implementation, and writing the initial draft.
**Mohamed El-Dosuky**: Technical supervision and editing the manuscript.
**Sherif Kamel**: Identification of the research problem and supervision.

## Ethics

This piece of writing is unique and includes unreleased content. All co-authors have read and approved the article, and the corresponding author attests that there are no ethical concerns.

## References

Acar, A., Aksu, H., Uluagac, A. S., & Conti, M. (2018). A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys, 51*(4), 1–35. https://doi.org/10.1145/3214303

Alloghani, M., Alani, M. M., Al-Jumeily, D., Baker, T., Mustafina, J., Hussain, A., & Aljaaf, A. J. (2019). A systematic review on the status and progress of homomorphic encryption technologies. *Journal of Information Security and Applications, 48*, 102362. https://doi.org/10.1016/j.jisa.2019.102362

Alon, B., Omri, E., & Paskin-Cherniavsky, A. (2020). MPC with friends and foes. In *Annual International Cryptology Conference* (pp. 677–706). Springer. https://doi.org/10.1007/978-3-030-56880-1_24

Aslanpour, M. S., Toosi, A. N., Cicconetti, C., Javadi, B., Sbarski, P., Taibi, D., & Dustdar, S. (2021). Serverless edge computing: Vision and challenges. In *Proceedings of the 2021 Australasian Computer Science Week Multiconference* (pp. 1–10). https://doi.org/10.1145/3437378.3444367

Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., & Shmatikov, V. (2020). How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics* (pp. 2938–2948). PMLR. https://doi.org/10.48550/arXiv.1807.00459

Bhagoji, A. N., Chakraborty, S., Mittal, P., & Calo, S. (2019). Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning* (pp. 634–643). PMLR. https://doi.org/10.48550/arXiv.1811.12470

Bouacida, N., & Mohapatra, P. (2021). Vulnerabilities in federated learning. *IEEE Access, 9*, 63229–63249. https://doi.org/10.1109/ACCESS.2021.3075203

Canetti, R. (2000). Security and composition of multiparty cryptographic protocols. *Journal of Cryptology, 13*, 143–202. https://doi.org/10.1007/s001459910006

Cerdeira, D., Santos, N., Fonseca, P., & Pinto, S. (2020). SoK: Understanding the prevailing security vulnerabilities in TrustZone-assisted TEE systems. In *2020 IEEE Symposium on Security and Privacy (SP)* (pp. 1416–1432). IEEE. https://doi.org/10.1109/SP40000.2020.00061

Chakrabarti, S., Knauth, T., Kuvaiskii, D., Steiner, M., & Vij, M. (2020). Trusted execution environment with Intel SGX. In *Responsible Genomic Data Sharing* (pp. 161–190). Academic Press. https://doi.org/10.1016/B978-0-12-816197-5.00008-5

El Ouadrhiri, A., & Abdelhadi, A. (2022). Differential privacy for deep and federated learning: A survey. *IEEE Access, 10*, 22359–22380. https://doi.org/10.1109/ACCESS.2022.3151670

Fang, M., Gong, N. Z., & Liu, J. (2020). Influence function based data poisoning attacks to top-N recommender systems. In *Proceedings of The Web Conference 2020* (pp. 3019–3025). https://doi.org/10.1145/3366423.3380072

Fereidooni, H., Marchal, S., Miettinen, M., Mirhoseini, A., Möllering, H., Nguyen, T. D., & Zeitouni, S. (2021). SAFELearn: Secure aggregation for private federated learning. In *2021 IEEE Security and Privacy Workshops (SPW)* (pp. 56–62). IEEE. https://doi.org/10.1109/SPW53761.2021.00017

Gafni, R., Aviv, I., & Haim, D. (2024). Multi-party secured collaboration architecture from cloud to edge. *Journal of Computer Information Systems, 64*(5), 698–709. https://doi.org/10.1080/08874417.2023.2248921

Gong, X., Chen, Y., Wang, Q., & Kong, W. (2022). Backdoor attacks and defenses in federated learning: State-of-the-art, taxonomy, and future directions. *IEEE Wireless Communications, 30*(2), 114–121. https://doi.org/10.1109/MWC.017.2100714

Jagielski, M., Oprea, A., Biggio, B., Liu, C., Nita-Rotaru, C., & Li, B. (2018). Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE Symposium on Security and Privacy (SP)* (pp. 19–35). IEEE. https://doi.org/10.1109/SP.2018.00057

Lim, W. Y. B., Luong, N. C., Hoang, D. T., Jiao, Y., Liang, Y. C., Yang, Q., & Miao, C. (2020). Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials, 22*(3), 2031–2063. https://doi.org/10.1109/COMST.2020.2986024

Lu, S., Yao, Y., & Shi, W. (2021). CLONE: Collaborative learning on the edges. *IEEE Internet of Things Journal, 8*(13), 10222–10236. https://doi.org/10.1109/JIOT.2020.3030278

Lyu, L., Yu, H., & Yang, Q. (2020). Threats to federated learning: A survey. *arXiv preprint*. https://doi.org/10.48550/arXiv.2003.02133

Ma, J., Naas, S. A., Sigg, S., & Lyu, X. (2022). Privacy-preserving federated learning based on multi-key homomorphic encryption. *International Journal of Intelligent Systems, 37*(9), 5880–5901. https://doi.org/10.1002/int.22818

Mitra, A., Biswas, S., Adhikari, T., Ghosh, A., De, S., & Karmakar, R. (2020). Emergence of edge computing: An advancement over cloud and fog. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pp. 1–7). IEEE. https://doi.org/10.1109/ICCCNT49239.2020.9225270

Muñoz-González, L., Biggio, B., Demontis, A., Paudice, A., Wongrassamee, V., Lupu, E. C., & Roli, F. (2017). Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security* (pp. 27–38). https://doi.org/10.1145/3128572.3140451

Nguyen, T. D., Marchal, S., Miettinen, M., Fereidooni, H., Asokan, N., & Sadeghi, A. R. (2019). DÏoT: A federated self-learning anomaly detection system for IoT. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)* (pp. 756–767). IEEE. https://doi.org/10.1109/ICDCS.2019.00080

Rieger, P., Nguyen, T. D., Miettinen, M., & Sadeghi, A. R. (2022). DeepSight: Mitigating backdoor attacks in federated learning through deep model inspection. *arXiv preprint arXiv:2201.00763*. https://doi.org/10.14722/ndss.2022.23156

Shafahi, A., Huang, W. R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., & Goldstein, T. (2018). Poison frogs! Targeted clean-label poisoning attacks on neural networks. In *Advances in Neural Information Processing Systems, 31*. https://doi.org/10.48550/arXiv.1804.00792

Shah, M., Zhang, W., Hu, H., & Yu, N. (2019). Paillier cryptosystem based mean value computation for encrypted domain image processing operations. *ACM Transactions on Multimedia Computing, Communications, and Applications, 15*(3), 1–21. https://doi.org/10.1145/3325194

Shokri, R., & Shmatikov, V. (2015). Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (pp. 1310–1321). https://doi.org/10.1145/2810103.2813687

Sun, Z., Wang, Y., Shu, M., Liu, R., & Zhao, H. (2019). Differential privacy for data and model publishing of medical data. *IEEE Access, 7*, 152103–152114. https://doi.org/10.1109/ACCESS.2019.2947295

Xie, C., Koyejo, S., & Gupta, I. (2019). Asynchronous federated optimization. *arXiv preprint arXiv:1903.03934*. https://doi.org/10.48550/arXiv.1903.03934

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. https://doi.org/10.1109/JIOT.2024.3382875

Zavalyshyn, I., Duarte, N. O., & Santos, N. (2018). HomePad: A privacy-aware smart hub for home environments. In *2018 IEEE/ACM Symposium on Edge Computing (SEC)* (pp. 58–73). IEEE. https://doi.org/10.1109/SEC.2018.00012

Zhang, B., Lu, G., Qiu, P., Gui, X., & Shi, Y. (2023). Advancing federated learning through verifiable computations and homomorphic encryption. *Entropy, 25*(11), 1550. https://doi.org/10.3390/e25111550

Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., & Gao, Y. (2021). A survey on federated learning. *Knowledge-Based Systems, 216*, 106775. https://doi.org/10.1016/j.knosys.2021.106775

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://doi.org/10.48550/arXiv.1806.00582

Zhou, X., Xu, M., Wu, Y., & Zheng, N. (2021). Deep model poisoning attack on federated learning. *Future Internet, 13*(3), 73. https://doi.org/10.3390/fi13030073

Zhu, T., Ye, D., Wang, W., Zhou, W., & Yu, P. S. (2020). More than privacy: Applying differential privacy in key areas of artificial intelligence. *IEEE Transactions on Knowledge and Data Engineering, 34*(6), 2824–2843. https://doi.org/10.1109/TKDE.2020.3014246