Securing Federated Learning: A Cryptographic Defense against Privacy Attack - MIA

Charishma Rathan Bala *R11908154*
Keerthiga Kalidas *R11903641*
Neha Bollu *R11903528*
Sruthi Mandalapu *R11906160*

I. Members contributions

- Keerthiga: Induce FL setup for client-server architecture
- Sruthi: Implementing MIA Attack by querying inputs
- Charishma: Applying of HE as defense
- Neha: Test the application after HE setup and produce analysis

II. Introduction

As computing devices become increasingly ubiquitous, people generate huge amounts of data through their day-to-day usage. Collecting such data into centralized systems raises a critical concern regarding data privacy and user confidentiality, as the data usually contains sensitive information (1). In this scenario, Federated Learning (FL), also well known as collaborative learning, which distributes model training to the devices from which data originate, emerged as a promising alternative ML paradigm (2). It is a collaboratively decentralized privacy-preserving technology to overcome challenges of data sensitivity (1). In machine learning terms, FL is a setting where many clients (e.g., mobile devices or whole organizations) collaboratively train a model under the orchestration of a central server (e.g., a service provider), while keeping the training data decentralized. It embodies the principles of focused data collection and minimization and can mitigate many of the systemic privacy risks (3).

Though federated learning mitigates data privacy, it is susceptible to model inversion attacks which is machine learning security threat. This attack uses the output of a model based on some of the parameters or the architecture of the model and attempts to reformulate the data (5). So, this MIA should be considered in federated learning as they pose significant privacy issues typically by accessing the model updates when a client communicates with the server. This pertains to data reconstruction, risking the leakage of sensitive information and eroding trust in the FL system (4).

A. *Motivation concerning privacy of data*

As existing FL lacks comprehensive defender's perspective, hence FL remains susceptible to MIA attack. In this paper we exhibit the risk for MIA attack and the defense mechanism. In order to address these challenges, we introduce Federated Cryptography Defense .This is a unified defense mechanism for Federated Learning (FL) designed to tackle one such privacy-centric attack which is model inversion attacks (MIA) (6). The defense system is built on Homomorphic Encryption (HE) to produce encrypted model updates to the server. The server combines the encrypted model updates and returns the aggregated encrypted updates back to the client. So, during this process attacker uses MIA attack, as we use homomorphic encryption it stays as robust defense, ensuring and preventing the attacker from formulation of data. We further look through the usage of defense mechanism in detail in further sections.

B. *Objectives concerning defense*

1) **Data Privacy:** Homomorphic encryption keeps client data confidential by encrypting model updates before sharing them. This ensures that even if intercepted, sensitive data remains secure and inaccessible. It minimizes privacy risks associated with unencrypted data transfer in federated learning.
2) **Defense Against Reconstruction Attacks:** Homomorphic encryption prevents attackers from reverse-engineering data from encrypted model updates. This defense protects sensitive information from attacks attempting to reconstruct original data. It strengthens federated learning's privacy against data inference threats.
3) **Secure Collaborative Learning:** Federated learning relies on collaboration among multiple clients to train a shared model effectively. But without encryption, sensitive data may unintentionally be inferred from model updates. This approach combines all client updates into a new model without revealing individual data, enabling secure collaboration where clients can contribute without risking data exposure.

C. *Contributions for homomorphic encryption*

This project presents a realistic threat model for model inversion attack that reflects potential adversarial scenarios. This model illustrates the vulnerabilities and impact of the proposed defense mechanism. The unified defense mechanism explains about homomorphic encryption (7), allows processing of encrypted data without decryption. It is prominent due to its ability to perform computations on encrypted data without requiring decryption, ensuring data privacy. This enables the clients to send encrypted updates for secure aggregation on a central server. So, this setup contributes a novel defense framework to federated learning, emphasizes the defender's perspective, and addresses privacy threats in Federated Learning (FL).

III. Related work

A. Security issue concerning FL

Federated learning (FL) offers a promising approach to decentralized machine learning, enabling model training across multiple clients while keeping data localized. Despite its advantages, federated learning faces several challenges. Here are a few scenarios explaining the vulnerabilities affecting FL system. The first scenario is when the attackers introduce adversarial examples during the training process, targeting weaknesses in the model architecture. These carefully manipulated inputs can cause the model to misclassify or produce incorrect outputs, posing significant risks in applications requiring high accuracy, such as security systems or healthcare diagnostics. Another type of vulnerability is Membership inference attacks which enable adversaries to determine whether specific data points were included in the model's training set. Attackers can analyze the model's predictions and confidence levels to make inferences about individual data points. MIA is also susceptible to exploiting the outputs of trained models to reconstruct sensitive input data. Attackers can query the model with carefully chosen inputs and analyze the returned confidence scores to infer information about the training data. This method allows for the gradual reconstruction of original data, such as facial images in recognition systems.

B. Exploring MIA Attack: Case Studies and Implications

As we discussed MIA attack and its vulnerability, now we demonstrate the previous studies explaining this attack.

Fredrikson et al. (2015) explains model inversion attacks in facial recognition systems, demonstrating how attackers could use confidence scores to reconstruct images of individuals from the training data. The researchers conducted experiments using a facial recognition model trained on images of individuals. They had successfully reconstructed facial images with high fidelity using only a few queries. And highlighted potential defenses, such as limiting information disclosure and implementing differential privacy techniques and Output regularization (8).

Zhu et al. (2019) extended the concept of model inversion attacks to the federated learning paradigm, where multiple clients collaboratively train a model without sharing their raw data. The authors explored how model inversion could still be a threat in this distributed setting, raising concerns about data privacy even when data remains local. They had introduced a comprehensive framework for conducting model inversion attacks specifically in federated learning scenarios. They detailed how attackers could exploit the model updates shared among clients to infer sensitive information about the local datasets. The attack utilized gradient updates sent from clients to the central server. By analyzing these updates, attackers could glean insights into the structure and content of the training data, reconstructing sensitive samples from individual clients. They have implemented their attack on a federated learning setup using datasets like MNIST. They demonstrated that, despite the decentralized nature of the learning process, attackers could successfully reconstruct significant portions of the original data by leveraging the gradients. To mitigate authors suggested differential privacy and output perturbation (9).

Shokri et al. (2017) expands on the concept of model inversion attacks, linking them to membership inference attacks. The researchers developed a framework to analyze the risk of model inversion and membership inference attacks on different deep learning architectures. They conducted experiments using various datasets, including images and text, to demonstrate the effectiveness of these attacks across different contexts. By using a series of model queries and observing the output confidence levels, they could reconstruct original data points. The study confirmed that model inversion attacks could successfully recover sensitive training inputs, especially in models that overfit their training data. Moreover, the attackers could leverage the model's confidence scores to improve their reconstruction accuracy, highlighting a correlation between model confidence and vulnerability to inversion attacks. To address these vulnerabilities, Shokri et al. recommended implementing robust privacy-preserving techniques such as differential privacy and secure aggregation (10).

C. Defence Approach: Homomorphic Encryption

Differential privacy is one such framework as explained in above scenarios as a common implication for protecting the data. It does carry out certain trade-offs when it aims to provide strong privacy guarantees by adding noise to the data, this noise can significantly degrade the utility of the data for analysis. When implementing differential privacy, the amount of noise added to ensure privacy is often inversely related to the accuracy of the results. So, we introduce Homomorphic Encryption, by incorporating HE into federated learning does not significantly compromise model accuracy. The proposed framework ensures that the training process is secure and privacy-preserving.

IV. Proposed Methodology

A. Threat Model

We explain about Threat model for MIA attack, here the attacker has access to the model, typically only able to query it with different inputs and receive output labels or confidence scores. Here, the attacker iteratively submits various inputs, analyzing the responses to deduce information about the target class. This process often involves a trial-and-error approach, where the attacker refines their inputs based on the feedback received, seeking to uncover details about the training data associated with the target output. As the attacker iteratively refines the inputs, their objective is to generate an input that yields a high confidence score for the selected target output. Below are the steps describing the threat model:

- Selecting a Target Class or Output – We have taken MNIST dataset, there are 10 classes
- Iteratively optimizing inputs – Grey, Black, White, Random Images

- Reconstruction or Approximation – Based on queried inputs,grey scale reconstructs to observe the image to identify it as a numbers dataset producing with less pixel quality (having similar reconstructions with that of MNIST)
- Finally, we test on average case inputting some numbers dataset (this is based on assumptions in previous step), the reconstructed images could be observed.

B. System assumptions/components:

1) **Clients:** Each client represents a data owner with access to local datasets. Clients train their models locally and send aggregated updates (model parameters) to the server.
2) **Central Server:** The server aggregates model updates from multiple clients to create a global model. It facilitates communication and coordinates the federated learning process.
3) **Attacker:** An external entity that aims to exploit vulnerabilities in the system. The attacker can intercept, analyze, and manipulate model updates sent from clients to the server.
4) **Local Models:** Models trained by individual clients using their local datasets. Each local model captures the specific characteristics of the data without exposing the data itself.
5) **Global Model:** The aggregated model produced by the server, which incorporates updates from all participating clients. This model aims to generalize well across different datasets.
6) **Communication Channel:** The network pathway through which clients send model updates to the server. This channel is a potential attack vector for intercepting model parameters.
7) **Inversion Model:**A model used by the attacker to infer or reconstruct sensitive information from the outputs of the target model. The inversion model leverages the intercepted parameters and outputs for data reconstruction.

Let $C = \{c_1, c_2, \ldots, c_n\}$ denote the set of n clients. Each client c_i has access to a local dataset D_i such that:

$$D_i = \{(x_j, y_j)\}_{j=1}^{m}$$

where x_j is the input data and y_j is the corresponding label. Each client trains a local model M_i using their dataset D_i and a local loss function L_i:

$$M_i = L_i(M, D_i)$$

Let θ_i denote the parameters of the local model M_i after training on client C_i. The client sends these parameters to the central server.

Global Model Aggregation

The server aggregates the model parameters from all clients to form a global model M_g:

$$\theta_g = \frac{1}{n} \sum_{i=1}^{n} \theta_i$$

Model Inversion Attack

An attacker, A, can intercept the model parameters θ_i during transmission. The attacker aims to reconstruct the original data D from the model parameters by using an inversion model to infer sensitive information.

The inversion model attempts to predict the input data x from the output y:

$$I(y) \rightarrow x'$$

where y could be class probabilities or model outputs related to D. The attacker seeks to minimize the reconstruction error E between the reconstructed input x' and the original input x:

$$E = |x - x'|$$

In a black-box setting, the attacker can query the model with various inputs and observe outputs:

$$y_j = P(M; x_j) \quad \forall j$$

Overall, the attacker A notices model parameters θ_i' sent from clients c_i to the server. Using an inversion model I, A reconstructs sensitive input data x by minimizing the reconstruction error E. The efficiency of the attack depends on the specific characteristics of the model M.

C. Approach for addressing Homomorphic Encryption

Introducing homomorphic encryption (HE) in federated learning is a powerful approach to secure data processing while preserving privacy. Below are the key steps to implement homomorphic encryption within a federated learning setup:

1) **Initialize Encryption Parameters**: Set parameters such as key size, security level, and encryption depth based on the HE schemes chosen.
2) **Encrypt Local Model Updates (Client-Side)**: After training a model on local data, each client encrypts its model updates (weights or gradients) using the homomorphic encryption scheme. This ensures that updates are encrypted before leaving the client's device, securing the data against interception.
3) **Send Encrypted Updates to the Server**: Clients send their encrypted updates to the central server. The server receives only encrypted data, so it has no access to the raw model updates, preserving client privacy.
4) **Aggregate Encrypted Updates (Server-Side)**: Using HE, the server performs aggregation operations (e.g., summing gradients) directly on the encrypted data. Since homomorphic encryption allows for computations on

encrypted data, this process maintains data security without requiring decryption.

5) **Decrypt Aggregated Model (Client-Side)**: Once aggregation is complete, the server sends the aggregated encrypted model update to a trusted client or a set of clients for decryption. Clients with decryption keys can decrypt the aggregated model update.
6) **Repeat FL Process**: This process is carried out through every round while the client is communicating with the server.

D. *Tools and Technologies*

We have used the Flower (`flwr`) framework in python and right now taking a basic CNN model to test on FL setup. We found few complications while using flower, so we were running the FL setup as an iterative setup. Right now, we are using CNN model, dataset is MNIST as a basic one. For implementing Model Inversion Attack – we are using ART.

V. Others

A. *Expected Challenges*

- **Setting up the FL to work on a single machine:** Federated Learning typically involves multiple clients with decentralized data. Simulating multiple clients on a single machine can be resource-intensive and requires creating isolated environments for each client.
- **Using pre-defined model:** We have considered vgg16 for effective results. This had increased in latency when performing the attacks while querying each input.
- **Performance Overheads:**Increased in latency as we need to utilize setting up 2 or more models demonstrating one for server and remaining for clients hence increases the latency. As the number of clients increases, the communication overhead and complexity of managing encryption keys and data processing will also rise. In this scenario, we would enable to access gpu to avoid latency and overhead.

B. *Resources*

- Model - CNN model.
- Dataset - MNIST.
- Attack Modules - Adversarial Robustness Toolbox - MI-Face.

C. *Timeline*

- Current progress – Built FL (with client-server architecture - using iterative loop by aggregating Fed-Avg) setup using CNN model using MNIST dataset, implemented the MIA attack.
- 11/10/2024 - Implement homomorphic encryption on FL setup and test for MIA attack for reconsrtucting the inputs.

References

[1] Li, L., Fan, Y., Tse, M., & Lin, K. (2020). A review of applications in federated learning. *Computers & Industrial Engineering*, 149, 106854.

[2] Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. (2021). Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2), 1–210.

[3] Zhou, Y., Ye, Q., & Lv, J. (2021). Communication-efficient federated learning with compensated overlap-fedavg. *IEEE Transactions on Parallel and Distributed Systems*, 33(1), 192–205. IEEE.

[4] Shi, S., Wang, N., Xiao, Y., Zhang, C., Shi, Y., Hou, Y. T., & Lou, W. (2023). Scale-mia: A scalable model inversion attack against secure federated learning via latent space reconstruction.

[5] Mohri, M., Sivek, G., & Suresh, A. T. (2019). Agnostic federated learning. In *International Conference on Machine Learning* (pp. 4615–4625). PMLR.

[6] Marcolla, C., Sucasas, V., Manzano, M., Bassoli, R., Fitzek, F. H. P., & Aaraj, N. (2022). Survey on fully homomorphic encryption, theory, and applications. *Proceedings of the IEEE*, 110(10), 1572–1609. IEEE.

[7] Marcolla, C., Sucasas, V., Manzano, M., Bassoli, R., Fitzek, F. H. P., & Aaraj, N. (2022). Survey on fully homomorphic encryption, theory, and applications. *Proceedings of the IEEE*, 110(10), 1572–1609.

[8] Fredrikson, M., Jha, S., & Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (pp. 1322–1333). ACM Press.

[9] Zhu, L., Liu, Z., & Han, S. (2019). Model inversion attacks against federated learning systems. In *Advances in Neural Information Processing Systems (NeurIPS)*. Accessed from NeurIPS proceedings.

[10] Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017). Membership inference attacks against machine learning models. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 3–18.