P. R. Ovi and A. Gangopadhyay, "A Comprehensive Study of Gradient Inversion Attacks in Federated Learning and Baseline Defense Strategies," 2023 57th Annual Conference on Information Sciences and Systems (CISS), Baltimore, MD, USA, 2023, pp. 1-6, doi: 10.1109/CISS56502.2023.10089719.

https://doi.org/10.1109/CISS56502.2023.10089719

# A Comprehensive Study of Gradient Inversion Attacks in Federated Learning and Baseline Defense Strategies

Pretom Roy Ovi, Aryya Gangopadhyay
Center for Real-time Distributed Sensing and Autonomy (CARDS)
University of Maryland, Baltimore County, USA
{povi1, gangopad}@umbc.edu

***Abstract*—With a greater emphasis on data confidentiality and legislation, collaborative machine learning algorithms are being developed to protect sensitive private data. Federated learning (FL) is the most popular of these methods, and FL enables collaborative model construction among a large number of users without the requirement for explicit data sharing. Because FL models are built in a distributed manner with gradient sharing protocol, they are vulnerable to "gradient inversion attacks," where sensitive training data is extracted from raw gradients. Gradient inversion attacks to reconstruct data are regarded as one of the wickedest privacy risks in FL, as attackers covertly spy gradient updates and backtrack from the gradients to obtain information about the raw data without compromising model training quality. Even without prior knowledge about the private data, the attacker can breach the secrecy and confidentiality of the training data via the intermediate gradients. Existing FL training protocol have been proven to exhibit vulnerabilities that can be exploited by adversaries both within and outside the system to compromise data privacy. Thus, it is critical to make FL system designers aware of the implications of future FL algorithm design on privacy preservation. Motivated by this, our work focuses on exploring the data confidentiality and integrity in FL, where we emphasize the intuitions, approaches, and fundamental assumptions used by the existing strategies of gradient inversion attacks to retrieve the data. Then we examine the limitations of different approaches and evaluate their qualitative performance in retrieving raw data. Furthermore, we assessed the effectiveness of baseline defense mechanisms against these attacks for robust privacy preservation in FL.**

***Index Terms*—Model inversion attacks, Gradient leakage attacks, Mixed quantization, Federated learning.**

## I. Introduction

Preservation of data security while building a deep learning model has always been a critical research problem. As a decentralized alternative to traditional ML model development, Federated Learning (FL) [1] is client-server based framework that facilitates collaborative training in the client side. With the advent of federated learning, research in this domain has received a significant acceleration. FL is mainly utilized in the areas where data confidentiality and information security is of the utmost importance e.g., in contested environment [2]. By design, the worker nodes (e.g., mobile devices) in federated learning are protected from data intrusion; they are able to keep their original data private on their devices while training a model in alongside with one another, and they only need to send their local model updates to the server. However, FL's default privacy is not sufficient to protect the confidentiality of local training data. If an adversary is able to intercept the local gradient update that a worker node shares with the FL server, it will be able to rebuild the local training data with high reconstruction accuracy. This gives the attacker a way to secretly spy on the private training data and leaves the FL system susceptible to privacy leakage attacks. Gradient inversion attacks is a type of privacy leakage attacks which allow attackers to recover private data by discreetly snooping on gradient adjustments during iterative training [3], [4]. The attacker can access the training data without having any prior knowledge of the learning model by exploiting the intermediate gradients.

Recently, privacy leakage attacks have been investigated extensively, and there are several such methods e.g., DLG [5], IDLG [6], GS [7], CPL [8], Grad Inversion [9], R-GAP [10], COPA [11], etc. These attacks can completely steal the training data and/or associated labels from gradients. Some of these attacks are iteration-based and others are recursion-based. The iteration-based attacks aim to minimize the distance between the dummy gradients and ground-truth gradients. Taking the distance between the gradients as error and the dummy inputs as parameters, the recovery process is formulated as an iterative optimization problem. DLG, iDLG, Grad Inversion are the attacks of iteration-based framework. On the other hand, recursion-based attacks [11]–[13] are also capable of reconstructing the original data in a closed-form algorithm. The key insight is to exploit the implicit relationships among the input data, model parameters, and gradients of each layer in order to find the optimal solution with the minimum error. R-GAP, COPA are the attacks of such types.

Some of the most explored prevention methods are Gaussian or Laplacian noise-based differential privacy (DP), gradient comprerssion, and homomorphic encryption (HE). In the first approach, to protect the confidentiality of training sample, Gaussian or Laplacian noise is added with gradients during training prior to share with the server [14], [15]. But the expense of accuracy may deteriorate below the threshold level.

The second is “gradient compression" such as pruning [5] and mixed quantization [16]. In gradient pruning, a specific pruning ratio is selected during training to make it robust against leakage attacks. But, pruning the network from the first epoch of training is not recommended as it may cause the loss of fundamental feature-related information. Another method of defending attacks is to use mixed precision and quantization. Here, each layer of the deep model is quantized into a different bit and because of different quantization in different layers, gradient optimization suffers to minimize the distance between dummy and ground truth gradients. The other one is homomorphic encryption [4], [17], where key-based encryption is proposed and often used in FL to protect the data privacy. While preserving the privacy of training data samples, HE theoretically ensures no performance loss in terms of model convergence [4], [17]. However, the effectiveness of HE comes at the expense of computation and memory, which limits its application. Apart from the above mentioned defense strategies, there are other ways to mitigate privacy leakage by increasing the local iterations or batch sizes [8], [18] during model training.

In this paper, we provide an overview of existing iteration-based and recursion-based gradient inversion attacks and compare their effectiveness in retrieving the training data. Through attack characterization, we provide a comprehensive understanding of the various attack mechanisms and attack surfaces that an adversary may exploit to reconstruct the private local training data from local model updates. For a comprehensive analysis of privacy threats, we present existing baseline mitigation strategies to defend the attacks and assess the maximum tolerance of attack methods against these defense strategies by attack effect analysis. We also compare how well these defense strategies work in a federated framework and what effect they have on the performance of global model.

## II. Federated Learning

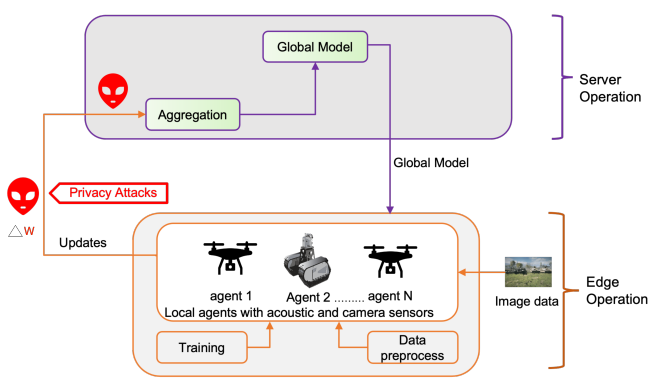



Fig. 1: Attack Surface in FL Framework.

In this section, we describe the detailed architecture and work flow of federated training set up based on FedAvg [1] algorithm where some of the notations used in this description are $\mathcal{N} = \{1, \ldots, N\}$ signify the set of $N$ clients, each of which has their own dataset $D_{k\in\mathcal{N}}$. Each of them trains a local model using their own dataset and only shares the model parameters with the FL server. Then, the global model, $\mathbf{w}_G$ formation takes place with all the local model updates which can denoted by $\mathbf{w} = \cup_{k\in\mathcal{N}}\mathbf{w}_k$. The attack surface of gradient inversion attacks in federated framework is depicted in figure 1. The process based on the workload of server and client is described below:

1) **Executed at the client level**
   - *Local training and updated parameter transmission:* Each client will be trained on local training samples to learn parameters after receiving the global model $\mathbf{w}_G^t$ from the server, where $t$ stands for each iteration index. For each client image data $D^i$ are fed to the local model for training. The client tries to minimize the loss function [19] $L\left(\mathbf{w}_k^t\right)$ and searches for optimal parameters $\mathbf{w}_k^t$.

$$\mathbf{w}_k^{t^*} = \arg\min_{\mathbf{w}_k^t} L\left(\mathbf{w}_k^t\right) \tag{1}$$

   After each round of training, updated local model parameters are sent to the server afterwards. In addition, each client will sent the last "Relu" layer activation value of local training samples that will be utilized to detect data level poisoning.

2) **Executed in server**
   - *Weight initialization:* The server determines the type of application and how the user will be trained. Based on the application, the global model is built in the server. The server then distributes the global model $\mathbf{w}_G^0$ to selected clients.
   - *Aggregation and global update:* The server aggregates the local models from the participants by discarding the detected compromised clients and then sends the updated global model parameters $\mathbf{w}_G^{t+1}$ back to the clients. The server wants to minimize the global loss function [19] $L\left(\mathbf{w}_G^t\right)$, i.e.

$$L\left(\mathbf{w}_G^t\right) = \frac{1}{N}\sum_{k=1}^{N} L\left(\mathbf{w}_k^t\right) \tag{2}$$

   This process is repeated until the global loss function converges or a desirable training accuracy is achieved. The Global Updater function runs on the SGD [20] formula for weight update. The formal equation of global loss minimization formula by the averaging aggregation at the $t^{th}$ iteration is given below:

$$\mathbf{w}_G^t = \frac{1}{\sum_{k\in\mathcal{N}} D_k}\sum_{k=1}^{N} D_k \mathbf{w}_i^t \tag{3}$$

## III. Gradient Inversion Attacks

In FL, server and clients exchange intermediate learning results such as models or gradient updates. Local gradient updates could be leaked during the transmission period and/or in the honest-but-curious server side, as shown in figure 1. And existing researches already established that an adversary

can launch gradient inversion attacks to retrieve the training samples if the local gradient updates are obtained.

Most of the existing gradient inversion attacks solve an optimization problem, as illustrated in Figure 2. After retrieving gradient updates $\Delta W$, the attacker generates dummy samples ($\hat{X}$,$\hat{y}$), then dummy samples are fed through the obtained model to get the dummy gradients, and next minimizes the distance between the real gradients $\Delta W$ and dummy gradients $\Delta W^{'}$. During the optimization process, the values of the dummy samples are optimized, so that the dummy samples approximate the training samples as optimization progresses. And this optimization typically relies on an objective given below:

$$\min_{\hat{X}} \text{Dist}\left(\frac{\partial \mathcal{L}(F(W,\hat{X}),\hat{y})}{\partial W}, \nabla W\right) + \text{Reg}(\hat{X},\hat{y}) \quad (4)$$

Here, $W$ is the current values of the trainable parameters of the attacked neural network. $F$ is the forward propagation function of the model. $L$ is the loss function of the model. $\hat{X}$ is the generated reconstruction samples and $\hat{y}$ is the labels of the samples. $\Delta W$ is the gradient update received from a client where, $\Delta W = \frac{\partial \mathcal{L}(F(W,\hat{X}),y)}{\partial W}$. $Dist$ is a distance function such as the L2 distance [5]and the cosine distance [7], which are the two most widely used distance functions in gradient inversion attacks. $Reg$ refers to regularization terms. For attacks on image classification tasks, additional regularization terms can be leveraged to generate more natural images. For example, total variation [7] is employed to reduce image noise, whereas clipping terms are utilized to prevent out-of-range values for a pixel. Finally, the objective is to minimize the sum of the distance and regularization terms for generated samples.

Here Both dummy samples $\hat{X}$ and dummy labels $\hat{y}$ are simultaneously optimized to retrieve the local training data (X,y). Recently, to reduce the complexity and avoid joint optimization problem, some analytical approaches [6], [9] propose to infer the labels before conducting the optimization by analyzing the distribution of the gradient tensor of the last fully connected layer. Label inference is important to improve the reconstruction quality of gradient inversion attacks. Pseudocode for iteration based gradient inversion attacks is shown in algorithm 1. The basic optimization strategy of attacks are described below.

**Algorithm 1:** Generalized Gradient Inversion Attacks

**Require:** F(X;W): Differentiable learning model, $W$: Model parameters, $\Delta W$: Gradients produced by private training datum (X,y), $N$: maximum number of iterations, $\mu$: learning rate.

**Ensure:**
($\hat{X}$,$\hat{y}$): Dummy datum and label.

$\hat{\mathbf{X}} \leftarrow \mathcal{N}(0,1)$ (Initialize the dummy datum.)
$\hat{\mathbf{y}} \leftarrow arg\,min_i(\Delta_i \text{W})$ (Extract ground truth label)
**for** *each iteration i from 1 to N* **do**
  $L_G \leftarrow \text{Dist}\left(\frac{\partial \mathcal{L}(F(W,\hat{X}),\hat{y})}{\partial W}, \nabla W\right) + \text{Reg}(\hat{X},\hat{y})$ (Calculate the loss.)
  $\hat{\mathbf{X}} \leftarrow \hat{\mathbf{X}} - \mu \Delta_{\hat{X}} L_G$ (Update dummy data to match gradients.)
**end**

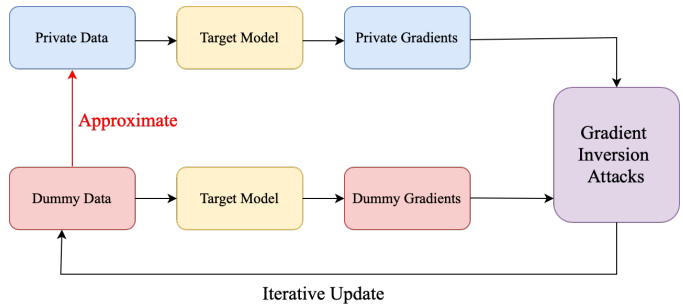



Fig. 2: Diagram of Gradient Inversion Attacks.

**DLG:** In deep leakage gradient, $\mathbf{L}_2$ distance between the private gradients and dummy gradients is considered to retrieve the training samples, no regularization term is used, as in eqn 5. DLG currently works for batch size up to 8 and image resolution up to 64x64. Increasing the batch size makes the leakage more difficult because it needs to solve more variables during optimization.

$$L_G = ||\Delta W^{'} - \Delta W||^2 \quad (5)$$

**iDLG:** In iDLG, the ground-truth labels from the shared gradients are extracted first, then the data is extracted more effectively based on correct labels which makes the training of iDLG easier to converge. iDLG always extracts the correct label as opposed to DLG which extracts wrong labels many times. iDLG can extract samples for batch size 1 and image resolution up to 32x32. iDLG also consider the $\mathbf{L}_2$ distance between the gradients as in eqn 5.

**GS:** In GS attacks, authors propose to use a cost function based on angles, i.e. cosine similarity and add total variation as regularization term. In comparison to euclidean distance, the objective is not to find images with a gradient that best fits the observed gradient, but to find images that lead to a similar change in model prediction as the ground truth. And optimization is based on eqn 6

$$L_G = 1 - cos(\Delta W^{'} - \Delta W) + \alpha TV(\hat{X}) \quad (6)$$

**CPL:** Client Privacy Leakage (CPL) attack is most effective when working with the gradient generated from the local training data of batch size 1. Furthermore, when the input data samples in a batch of size larger than one belong to only one or two classes, the CPL attacks can effectively reconstruct the training data of the entire batch. CPL attack can handle image resolution of size of $32 \times 32$, $64 \times 64$, and $128 \times 128$. It requires a much larger number of attack iterations in order to succeed the attack with high reconstruction performance. CPL also utilizes $\mathbf{L}_2$ distance between the gradients in addition with a label-based regularizer to improve the stability of the attack optimization, as in eqn 7.

$$L_G = ||\Delta W^{'} - \Delta W||^2 + \alpha ||F(\hat{X},W) - \hat{y}||^2 \quad (7)$$

**Grad-Inversion:** To recover more realistic data from the batch, some auxiliary regularization terms are incorporated into the cost functions, as in eqn 8. These regularizers are classified into two types- one that constrains the fidelity of images, and the other that revises the position of the main object. Fidelity regularization steers the reconstructed data from impractical images, including total variation norm (TV), $\mathbf{L}_2$ norm and batch normalization (BN). Group consistency regularization jointly considers multiple random seeds for initialization, and calculates the averaged data as reference.

$$L_G = ||\Delta W^{'} - \Delta W||^2 + \mathbb{R}_{fidelity}(\hat{X}) + \mathbb{R}_{group}(\hat{X}) \quad (8)$$

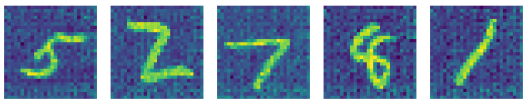
**R-GAP:** R-GAP is a recursion based gradient inversion attack. R-GAP recursively reconstructs each layer's input from the last layer to first by solving linear equations, which is limited to a batch size equal to 1. Here, the key insight is to exploit the implicit relationships among the input data, model parameters, and gradients of each layer in order to find the optimal solution with the minimum error.

## IV. Experimental Setup and Result Analysis

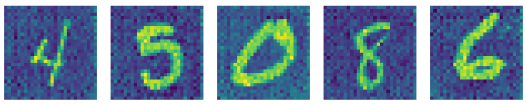
This section will step through the detailed experiment setup and result analysis. Experiments are conducted on a system with an Intel(R) 8 core Core(TM) i9-11900K CPU and an NVIDIA GeForce RTX 3090 GPUs.

**Dataset:** We utilized MNIST, FASHION-MNIST, and CIFAR10 dataset to retrieve the training samples and evaluate the performance of defense strategies.

**Gradient Inversion Attack Methods:** We utilized the LeNet-5 CNN architecture as the global target model. In this work, the gradient inversion attacks such as DLG, iDLG, GS, CPL, Grad Inversion and R-GAP were utilized to retrieve the training samples. And extracted samples from the attacks were depicted in figure 3 and figure 4.



(a) Extracted by Grad Inversion Attacks

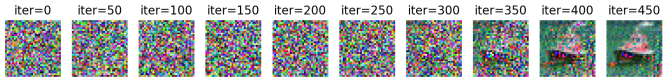


(b) Extracted by CPL Attacks

Fig. 3: Gradient Inversion Attacks on batch data.

**Defense Methods:** Here we experimented with baseline defense strategies: Mixed precision and quantization [16], Gradient Pruning and Differential Privacy (DP). DP protects privacy with theoretical guarantee by injecting noise to the gradients. Due to the fact that the majority of DPSGD implementations for image classification tasks employ a pre-training and fine-tuning pipeline, it is difficult to fairly compare them to other defense techniques that can directly apply when training the model from scratch. Thus, we separately apply


(a) Extracted by Deep Leakage Gradient (DLG)




(b) Extracted by Inverting Gradient (GS) attacks

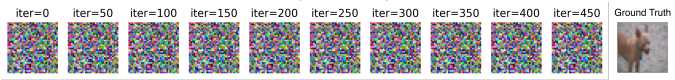
Fig. 4: Gradient Inversion Attacks on single training sample.

Gaussian and Laplacian noise to develop two DP baselines, namely DP-Gaussian and DP-Laplacian.

We examined and discussed the efficacy of these defensive measures below:

- *Mixed Precision and Quantization:* In mixed precision and quantization, each layer of deep model is quantized with different bits before sharing the local model updates to server. According to this study [16], authors utilized the combination of int8 and int16 quantization bit to quantize the gradients. Figure 5 depicted the retrieved training samples from gradients. In figure 5b, training images were not retrieved from the quantized gradients even after 450 iterations of distance minimization, whereas training samples were retrieved within 40 iterations when gradients are not quantized, shown in figure 5a.


(a) Leakage from gradients

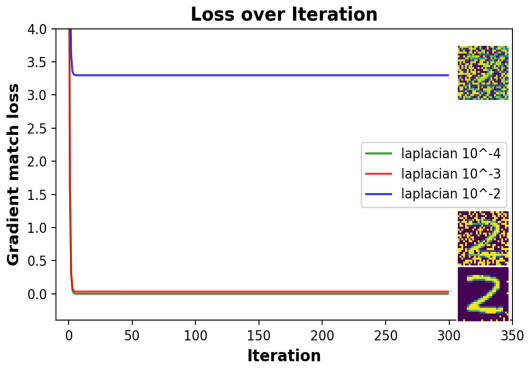



(b) No leakage after mixed precision applied

Fig. 5: Gradient Inversion attack. (a) Training sample retrieved within 40 iterations, (b) no leakage from gradients after mixed precision applied.

- *DP-Gaussian and DP-Laplacian:* To evaluate, we experimented with Gaussian and Laplacian noise variance range from $10^{-1}$ to $10^{-4}$ with mean 0. Figure 6 depicts the impact of varying noise levels against the leakage attack. When variance is at the scale of $10^{-4}$ or $10^{-3}$, the noisy gradients do not prevent the attack. For noise with variance level $10^{-2}$, though with artifacts, the leakage can still be performed. Furthermore, noise with variance larger than $10^{-2}$ can defend attacks. To defend against the leakage attack, Gaussian/Laplacian noise with a minimum variance level of $10^{-1}$ should be added to the gradients. In addition, we empirically tested this level of noise against all existing gradient inversion attacks and found that it can prevent them. However, this amount of noise has a significant degradation in performance.
- *Gradient Pruning:* Gradient Pruning prunes gradients that are smaller than a certain threshold. In figure 7, it is


(a) Different magnitude Gaussian noise




(b) Different magnitude Laplacian noise

Fig. 6: Effect of Gaussian and Laplacian noise against attacks.

observed that pruning in the range of 1% to 10% cannot defend the attacks because the reconstructed image reveals the data and is easily recognizable. For gradients with 20% pruning, though with artifacts, the gradient inversion attack is still successful. But when 30% pruning is applied on gradients, the attack is prevented for cifar-10 images. As we know recovering monochromatic images with a clean background (MNIST) is easier, and so it requires around 60% pruning to defend the attacks in retrieving mnist training samples. So the pixel complexity of training samples determines the required pruning rate to defend the attacks. For simple images, we need higher pruning rate.

### A. Defense Result Analysis

From the above result, we found out that Gaussian and Laplacian noise with variance level $10^{-1}$ can defend the gradient inversion attacks. Secondly, gradients with above 30% pruning for Cifar-10 data can prevent the leakage attacks. For the complex training samples, we need less pruning percentage whereas for monochromatic images with a clean background (e.g. MNIST), it requires higher pruning rate to prevent the attacks. So, we need to set perfect pruning rate by experimenting in an iterative way. However, the basic framework for pruning is to train the network first, then prune the less important part of network by setting it to 0, and

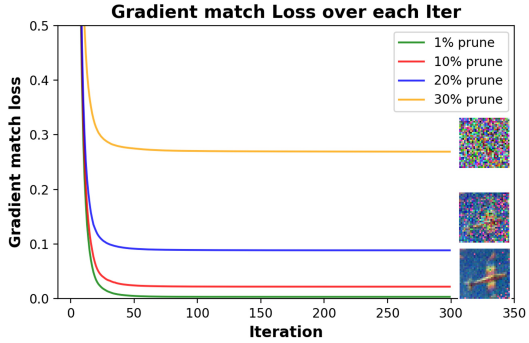



Fig. 7: Effect of Pruning against gradient leakage attacks.

TABLE I: Accuracy comparison of baseline defense strategies

| Methods | Dataset | | |
|---|---|---|---|
| | Mnist | Fashion Mnist | Cifar10 |
| Base FL (vulnerable to attack) | 97.05 | 86.8 | 60.04 |
| DP-Gaussian (variance=$10^{-1}$) | 85.54 | 73.8 | 42.64 |
| DP-Laplacian (variance=$10^{-1}$) | 78.66 | 66.19 | 33.41 |
| Mixed Precision and Quantization | **96.67** | **85.22** | **58.9** |

finally fine-tuning the network. Pruning removes the neuron that is the least significant. But in the first training round, the model cannot determine which neurons are significant and which are not. So, pruning the network from the first epoch of training is not recommended and may result in poor training performance. Moreover, convergence is an issue in federated training and clients may have local training data in an imbalanced fashion with varying class distributions. Therefore, pruning the network during training, particularly in a federated setup, may result in no convergence at all. Thirdly, in mixed quantization approach, each layer of any deep model can be quantized with different bits. Because of varying bit size in each layer, the existing gradient inversion attacks can not minimize the distance between private quantized gradients and dummy gradients and thus it is robust against the attacks. Besides defending the attack, authors of [16] suggested to dequantize the gradients in the server after/before aggregation to achieve desired performance in FL. And to dequantize the gradients and get the ground truth values, the set of operations that was used to quantize, the same set of reverse operations is needed. So, inherent limitation of this approach is that the server needs to know the operations that were used on the client side during quantization.

As stated earlier, recursion-based gradients inversion algorithms e.g., R-GAP, COPA does not minimize the distance between the dummy and ground truth gradients. Rather these attacks exploit the implicit relationships among the input data, model parameters, and gradients of each layer in order to find the optimal solution with the minimum error. Thus, defending such attacks is easier than the iterative optimization based attacks. To defend, adding a small amount of random noise with gradients is enough. Even pruning rate of only 1% can defend the recursion-based attacks.

Finally, we compared the performance accuracy of all the

baseline strategies by incorporating them in FL framework, results shown in table I. The server-clients-based federated learning framework is implemented on Keras, which is built on the TensorFlow backend. We utilized the LeNet-5 CNN architecture as a global model and simulated FL training with 15 clients. For each round, 1 epoch of local training is conducted on the client side. We used SGD optimizer and set the learning rate $\eta$ to 0.01 during training. To compare the effect of each baseline defense strategy on the performance, we reported the performance of FL utilizing the DP-Gaussian and DP-Laplacian with variance level $10^{-1}$ because this amount of noise is the least to prevent the attacks. And we determined that incorporating the DP-Gaussian and DP-Laplacian with variance level $10^{-1}$ in FL framework deteriorates the performance of global model. Then, we evaluated the performance of mixed quantization approach and found out that it can keep the desired accuracy level in FL while defending the attacks because it is possible to dequantize the gradients in the server end after/before aggregation. Since dequantization is not a fully reversible operation, we will not get the exact ground truth gradients and so, the accuracy will decrease slightly, but it is still superior to others approaches. In comparison with base FL (model with no defense against attacks), mixed precision and quantization achieved almost the same level of performance-only $0.5\%$ drop for MNIST and FASHION MNIST, and $1\%$ drop for CIFAR10, demonstrated in table I. On the other hand, DP-Gaussian has on average 15% drop and for DP-Laplacian 20% on average.

## V. Conclusion

In this paper, we explore the data confidentiality and model integrity in federated learning, where we emphasize the intuitions, approaches, and fundamental assumptions used by state-of-the-art gradient inversion attacks. We demonstrate how adversaries can reconstruct the private local training data from the shared parameter updates (e.g., gradient or weight updates) by launching different attack methods. Then we evaluate the performance of existing defense approaches against the attacks and assess the maximum tolerance of different attack methods against these defense strategies. Finally, we examine the effectiveness of defense mechanisms in federated framework and their impact on the performance of the global model.

## ACKNOWLEDGMENT

This research is partially supported by NSF Grant No. 1923982 and U.S. Army Grant No. W911NF21-20076.

## References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[2] P. R. Ovi, E. Dey, N. Roy, A. Gangopadhyay, and R. F. Erbacher, "Towards developing a data security aware federated training framework in multi-modal contested environments," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications IV*, vol. 12113. SPIE, 2022, pp. 189–198.

[3] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys (Csur)*, vol. 51, no. 4, pp. 1–35, 2018.

[4] Y. Aono, T. Hayashi, L. Wang, S. Moriai *et al.*, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2017.

[5] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Advances in neural information processing systems*, vol. 32, 2019.

[6] B. Zhao, K. R. Mopuri, and H. Bilen, "idlg: Improved deep leakage from gradients," *arXiv preprint arXiv:2001.02610*, 2020.

[7] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?" *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 937–16 947, 2020.

[8] W. Wei, L. Liu, M. Loper, K.-H. Chow, M. E. Gursoy, S. Truex, and Y. Wu, "A framework for evaluating gradient leakage attacks in federated learning," *arXiv preprint arXiv:2004.10397*, 2020.

[9] H. Yin, A. Mallya, A. Vahdat, J. M. Alvarez, J. Kautz, and P. Molchanov, "See through gradients: Image batch recovery via gradinversion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 337–16 346.

[10] J. Zhu and M. Blaschko, "R-gap: Recursive gradient attack on privacy," *arXiv preprint arXiv:2010.07733*, 2020.

[11] C. Chen and N. D. Campbell, "Understanding training-data leakage from gradients in neural networks for image classification," *arXiv preprint arXiv:2111.10178*, 2021.

[12] C. Fan and P. Liu, "Federated generative adversarial learning," in *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*. Springer, 2020, pp. 3–15.

[13] J. Zhu and M. Blaschko, "Differentially private sgd with sparse gradients," *arXiv preprint arXiv:2112.00845*, 2021.

[14] W. Wei, L. Liu, Y. Wut, G. Su, and A. Iyengar, "Gradient-leakage resilient federated learning," in *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2021, pp. 797–807.

[15] M. Naseri, J. Hayes, and E. De Cristofaro, "Local and central differential privacy for robustness and privacy in federated learning," *arXiv preprint arXiv:2009.03561*, 2020.

[16] P. R. Ovi, E. Dey, N. Roy, and A. Gangopadhyay, "Mixed precision quantization to tackle gradient leakage attacks in federated learning," *arXiv preprint arXiv:2210.13457*, 2022.

[17] H. Fang and Q. Qian, "Privacy preserving machine learning with homomorphic encryption and federated learning," *Future Internet*, vol. 13, no. 4, p. 94, 2021.

[18] Y. Huang, S. Gupta, Z. Song, K. Li, and S. Arora, "Evaluating gradient inversion attacks and defenses in federated learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 7232–7241, 2021.

[19] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[20] Z. Li, V. Sharma, and S. P. Mohanty, "Preserving data privacy via federated learning: Challenges and solutions," *IEEE Consumer Electronics Magazine*, vol. 9, no. 3, pp. 8–16, 2020.