

Protect Privacy from Gradient Leakage Attack in Federated Learning

Junxiao Wang[†], Song Guo^{†*}, Xin Xie^{†*}, Heng Qi[‡]

[†]Hong Kong Polytechnic University, Hong Kong, China

[‡]Dalian University of Technology, Dalian, China

{junxiao.wang, song.guo, xin-ryan.xie}@polyu.edu.hk[†], hengqi@dlut.edu.cn[‡]

*Corresponding authors: Xin Xie, Song Guo

Abstract—Federated Learning (FL) is susceptible to gradient leakage attacks, as recent studies show the feasibility of obtaining private training data on clients from publicly shared gradients. Existing work solves this problem by incorporating a series of privacy protection mechanisms, such as **homomorphic encryption and local differential privacy** to prevent data leakage. However, these solutions either incur significant **communication and computation costs**, or significant **training accuracy loss**. In this paper, we show that the sensitivity of gradient changes *w.r.t.* training data is an essential measure of information leakage risk. Based on this observation, we present a novel defense, whose intuition is perturbing gradients to match information leakage risk such that the defense overhead is lightweight while privacy protection is adequate. Our another key observation is that **global correlations of gradients could compensate for this perturbation**. Based on such compensation, training can achieve guaranteed accuracy. We conduct experiments on MNIST, Fashion-MNIST and CIFAR-10 for defending against two gradient leakage attacks. Without sacrificing accuracy, the results demonstrate that our lightweight defense can decrease the PSNR and SSIM between the reconstructed images and raw images by up to more than 60% for both two attacks, compared with baseline defensive methods.

I. INTRODUCTION

Federated Learning (FL) is an emerging distributed machine learning framework that enables a number of clients to train a shared model in a collaborative fashion without explicitly sharing their private data [1, 2]. A federated server coordinates the FL process, where each participating client communicates only the model or gradients with the federated server. Therefore, FL has been a natural choice for those mobile machine learning applications needing privacy protection, *e.g.*, speaker verification [3] and keyboard prediction [4].

Although FL provides default privacy by allowing clients to keep their private data on local devices, recent studies have noted that **shared gradients still leak private information**. The honest-but-curious server could reveal the privacy illegally and silently [5], by intercepting the gradients shared from clients and conducting gradient leakage attacks [6–8] to reconstruct the raw data (see II-B on Threat Model for detail).

To protect privacy of FL, the community consensus is to integrate privacy-preserving solutions with FL framework. Thus, an ideal privacy-preserving solution is expected to achieve: 1) **lightweight** in terms of computation, memory, and bandwidth, independent of the scale of clients. 2) **guaranteed** in terms of

training accuracy: a solution should realize a desirable privacy-performance tradeoff without the price of significant training accuracy loss. 3) **adequate** in terms of the privacy protection, which we define as the ability to capture the risk of information leakage and conduct targeted full defense.

Although existing studies have made notable contributions, they can hardly achieve the above three criteria simultaneously. We coarsely characterize them into four categories:

- **Homomorphic Encryption (HE)** solutions [9–11] allow certain computation (*e.g.*, addition) to be performed directly on encrypted gradients without decrypting them first. HE protects gradients from being exposed to any external parties including the server, as gradient aggregation is performed on ciphertexts. HE incurs no training accuracy loss, as no noise is added to gradients during encryption, while HE introduces significant overhead to computation and communication.
- **Local Differential Privacy (LDP)** solutions [12–15] allow clients to perform a differentially private transformation to their gradients, and ensure that each individual data sample from local datasets hard to be identified from the gradients, by injecting artificial noise into gradients at the client side. LDP entails **tradeoff between convergence performance and privacy protection**, however it means that stronger privacy protection leads to significant training accuracy loss.
- **Multi-Party Computation (MPC)** solutions [16, 17] allow multiple parties to collaboratively compute a function in a protocol that each party knows nothing except its input and output (*i.e.*, meeting zero-knowledge) of shared gradients. Their main bottleneck is the high communication cost, and the communication-efficient implementations require either extensive offline computation or lower privacy protection.
- **Targeted Defense (TD)** solution [18] is proposed recently, and specialized for a particular attack domain (*i.e.*, gradient leakage attacks in FL). Compared to above general-purpose solutions which can be broadly applicable across attack domains, TD closely fits characteristics of gradient leakage attack to lower its overhead, while this makes itself defense pattern rigid and easy to be inferred. Thus in practice, it fails short in terms of privacy protection (see II-D for detail).

In summary, no existing studies can achieve adequate defense without considerable defense overhead, and none focuses on lightweight and guaranteed defensive mechanism against the gradient leakage attack in FL.

We design a new defensive mechanism to meet the above three criteria simultaneously. In this research, we show our observation that the sensitivity of gradient changes *w.r.t.* **input data is an essential measure of information leakage risk**. Based on this observation, we propose a defense against gradient leakage attacks. The intuition of our defense is perturbing the gradients to match their leakage risk such that our defense overhead is lightweight while adequate privacy protection is maintained. We also show another significant observation that using **global correlations of gradients could compensate for this perturbation**. Based on such compensation, training can achieve guaranteed accuracy.

To support above, a new defensive mechanism incorporates two key techniques regarded as our main contributions:

- **Layer-wise information leakage risk quantification and balanced perturbation:** We move towards a more essential understanding of layer-wise information leakage from the gradients, using the *sensitivity* of gradient changes *w.r.t.* the **input information to quantify the leakage risk**. The easy-to-compute metric, sensitivity, can capture how information leaks in a layer-wise way, and facilitate **an effective design of perturbation** with full defense balanced across the layers. Such that the quality of the reconstructed data is severely degraded, while a lightweight defense is maintained.
 - **Perturbation compensation based on correlation among layer-wise gradient attributes:** The attributes of the global gradients have strong correlation because they are produced in backward propagation layer-wisely basing on a number of data samples of participants. The compensation is to exploit such a relationship, then based on which to derive, from the perturbed data, more accurate information about the gradient attributes, thereby reducing the footprints of perturbation and maintaining the guaranteed convergence performance.
- We also conduct large-scale simulations of image classification tasks in FL settings using most generalized deep networks (*i.e.*, convolutional neural networks) and popular datasets.
- **Experimental results validate our superior on preserving privacy and performance:** We evaluate proposed defensive mechanism against two gradient leakage attacks under the non-IID settings of FL, and show the results on three image datasets: MNIST, Fashion-MNIST, CIFAR-10. Overall seen from experimental results, our defensive mechanism could decrease the PSNR, SSIM between the reconstructed images and the raw images by as much as more than 60% for both two attacks, compared with the baseline defensive methods. Moreover, the results also validate that our mechanism can achieve such adequate defense with less than 2% accuracy loss and less than 10% training overhead increment.

II. PRELIMINARIES

For sake of completeness, we now briefly review the federated learning, correspondent threat model and the gradient leakage attacks introduced in [1, 5–8, 19]. We also discuss the targeted defense suggested in [18] for hiding the raw data (*i.e.*, protection against the gradient leakage attack) while still being able to expose privacy to attackers.

A. Federated Learning

The term *federated learning* was introduced by McMahan et al. [1]. Federated learning (FL) is machine learning setting where a set of n clients (*e.g.*, mobile devices) collaboratively train a model under the orchestration of a federated server (*e.g.*, service provider), while **the training data of clients is stored locally and not exchanged** [20]. The federated server orchestrates the collaborative training process, by repeating the following steps until training is converged:

1) **Client selection.** Given the unstable client availability, for the round t of federated learning, the federated server samples a small subset of m clients meeting eligibility requirements out of all n clients to participate in the learning.

2) **Local training.** Upon notification of being selected at the round t , each selected client downloads the current parameters θ of global model and a training program from the federated server. Each selected client locally computes an update to the global model on its local training data by executing the training program. More specifically, the gradients updated at one client (denoted as G), are computed by $\frac{\partial \ell(X, y, \theta)}{\partial \theta}$, where X, y denote the batches of training data and corresponding labels, and $\ell(\cdot)$ refers to the loss function.

The gradients G in typical federated learning settings are the minimum that must be shared to the server, corresponding to FedSGD approach. In FedAvg [1], models are consecutively updated on more batches of local data, which can be several epochs of training, and then shared. We note that a common way is to share the updated model $\theta + G$, but this practically amounts to sharing G since all participants know θ .

3) **Global Aggregation.** Upon having received local updates from m clients, the federated server aggregates these updates and update its global model, and initiates next round learning.



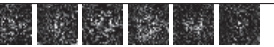
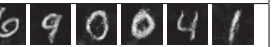


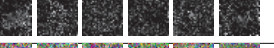

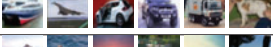
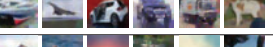
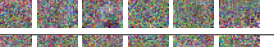
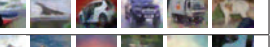
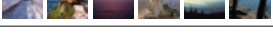



B. Threat Model

In federated learning, each individual client is susceptible to gradient leakage attacks because each client is the participant sending itself gradients to the federated server. We assume that the **federated server is an honest-but-curious server** with the goal of uncovering the local training data of clients. Specially, the federated server will honestly perform the aggregation of local gradients and manage the iteration rounds for federated learning. However, the federated server may be curious and may analyze periodic gradients from certain clients to perform the gradient leakage attacks and gain access to the training data of the victim clients. It is worth noting that even if the network connection between the client and federated server is secure, the gradient leakage attack can take place.

C. Gradient Leakage Attack

The gradient leakage attack [6–8] is a data reconstruction attack, in which the attacker designs a gradient-based reconstruction learning algorithm that will take the local gradients at round t , says G , to be shared by the client, to reconstruct the private data used in the local training. The gradient leakage attack works by randomly initializing dummy data and feeding it into the model to get dummy gradients. Then, the dummy

TABLE I
RAW DATA AND RECONSTRUCTED RESULTS OF [18] OVER MNIST, FASHION-MNIST, CIFAR-10 AND CIFAR-100 DATASETS.

Dataset	[A] Raw data	[B] Unperturbed	[C] Perturbed	[D] Muted
MNIST				
FASHION				
CIFAR-10				
CIFAR-100				

data gets close to the real private data if the dummy gradients are optimized to get close to the real gradients. The gradient leakage attack can reach high accuracy (e.g., to a pixel-wise level for images) as shown in experiments [6, 8].

D. Motivation

Some defensive mechanisms have been presented to protect privacy and can be categorized into three types: local differential privacy [12–15], secure multi-party computation [16, 17], and homomorphic encryption [9–11] but these mechanisms incur either unacceptable computational overheads or significant accuracy degrade. The reason is that these mechanisms are designed for general-purpose defense, instead specially for the gradient leakage attack.

The latest study [18] provides another viewpoint we name it the targeted defense. Its defense fits more closely to characteristics of gradient leakage attacks in FL to lower its footprints of defense. Specially, it only perturbs a certain single layer of the shared gradients, leading its defense pattern rigid and easy to be inferred, thus it fails short in privacy protection. More intuitively, we show its reconstructed results over 4 datasets.

As shown in Table I, column A is raw data of local training. Column B is the reconstructed results using the unperturbed gradients. Column C is results with gradients being perturbed by [18]. Column D is the results with the perturbed layer being muted by the attacker, i.e., the attacker only uses the rest of unperturbed layers to reconstruct data.

Although the approach suggested in [18] seems to provide protection under such an assumption: the attacker will naively use entire gradients for reconstruction. However, the defense was found in vain once the attacker mutes this certain layer, while remaining gradients without perturbation can still reveal privacy of raw data. This drawback motivates us to design a defensive mechanism that is hard to break by the attacker.

III. OVERVIEW OF DEFENSIVE MECHANISM

As shown in Figure 1, the workflow of proposed defensive mechanism consists of two components, called *local random perturbation* and *global update compensation*. Local random perturbation is performed on each individual client, in order to give a lightweight and adequate protection for the gradients shared from clients to the server (i.e., potential attacker). At the server, global update compensation is performed, in order to lower footprints of perturbation and training accuracy loss, thus to achieve guaranteed convergence performance.

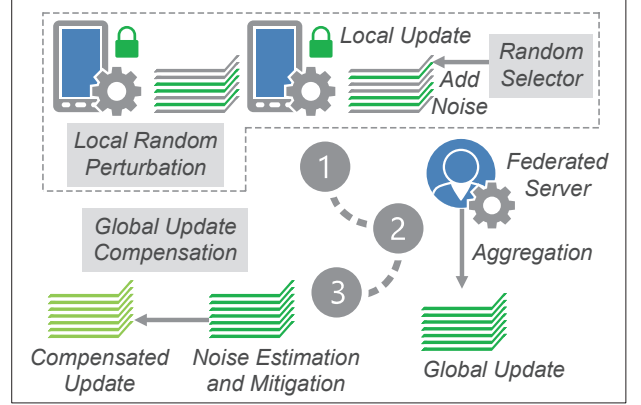


Fig. 1. Overview and workflow of the proposed defensive mechanism.

1) Shield training data with local random perturbation.

The attacks on gradients are proved to reveal privacy, but more fundamentally, it is not clear where and to what degree the private information are stored in the gradients. Our defense firstly uses the *sensitivity of gradient changes* w.r.t. input information to quantify the leakage risk in layer-wise manner. The easy-to-compute metric, sensitivity, can capture how information leaks in layer-wise manner, and facilitates the slicing of the gradients into multiple slices, each of which has balanced information leakage risk.

Then, in each round of learning, each client will randomly select a certain proportion of slices from its shared gradients and add artificial noise to these slices. Since all clients select their noise-adding slices locally and randomly, it is difficult for the attacker to locate these noisy slices, then it is difficult to mute these slices and perform data reconstruction. Upon the proportion of noisy slices in the gradients exceeds a certain of level, even if the attacker may locate the noisy slices by brute force, the attacker will not be able to reconstruct the data.

2) Retain convergence performance with global update compensation.

The layer-wise attributes of the global gradients are highly correlated because they are layer-wisely produced in the backward propagation basing on a number of data samples of participants. Since global gradients have strong correlations among its attributes, they have large variances in the directions of some vectors but small variances in the other directions. The addition of noise does not change the trends too much, because if the random variables added to the original

data are independent, their variances will be evenly distributed among all the directions.

Following this relationship, the compensation could derive, from the perturbed data, more accurate information about the global gradients, then based on which to reduce the footprints of perturbation and preserve the convergence performance.

3) [Optional] Prevent compensation from being abused with local clipping operation. Considering that the attacker may abuse compensation to upgrade the reconstructed results, *i.e.*, the attacker may use compensation to mitigate the noise in shared gradients and then perform reconstruction. We target this concern with the *local clipping operation*, by **clipping the gradients of individual data samples and scaling them to the similar range corresponding to the noise**. At the server, it is hard for the clipped gradients even if being compensated to reveal whether a particular data sample has participated in the learning. Instead, the global compensation is still valid because the correlation information accumulated in the global gradients comes from multiple participants and is based on a number of data samples (*i.e.*, redundant), which will clearly depict the strong correlation of the global gradients.

IV. LOCAL RANDOM PERTURBATION

A. Quantifying the Information Leakage Risk

Attacks on gradients are shown to be effective in practice [6–8], but more fundamentally, it is not clear where and to what degree the private information are stored in the gradients. Existing research considers information leakage to be a result of insufficient generalization performance or unintended memorization [21, 22]. Prior study [6] computed the mean square error loss between the dummy and real gradients layer-wisely, showing that the last one layer has the smallest loss. However, this result is not suitable for comparison across layers since the calculation of losses depends on layer sizes which usually are different among layers. Besides, the smallest loss does not necessarily result in the highest attack performance, and thus, the loss cannot unveil the essential cause of leakage.

Understanding the private information memorization, can be related to the model generalization ability: learning a model that generalizes well avoids the memorization of unnecessary information making its parameters insensitive to small changes in the inputs [23]. Based on this, if certain gradients are non-sensitive to changes in the input data, reconstructing the input will be more challenging, which makes the success rate of the attack low. The sensitivity *w.r.t.* the input data is such a concept that quantifies to what extent **the output is affected by small changes in the corresponding input data**, and has been used in measuring model generalization [24–26].

We adapt *sensitivity* to quantify the information leakage risk from the gradients. Let X denote a small training dataset (called *root* dataset) public between participants. This dataset may **originate from a publicly available data source**, a separate dataset from the client data which is not privacy sensitive, or perhaps a distillation of the raw data following [27]. Then, the information leakage risk is measured by **how much information about X an attacker could extract**. More specifically,

Algorithm 1: Local Random Perturbation

Input: G local gradients computed in one client;
 γ perturbation ratio predefined among participants;
 σ standard deviation of noise variables;
Output: G^* local gradients shared to the server;

```

1 Partition( $G$ )  $\rightarrow$  sliced  $G[0 \dots s-1]$ ;
2 Max( $1, \lfloor s \times \gamma \rfloor$ )  $\rightarrow$  perturbed slice number  $d$ ;
3 index = Slice_selector( $s, d$ );
4 for each  $i$  in index do
5    $G[i] = G[i] + \text{Gaussian}(0, \sigma^2)$ ;
6 return  $G^* = G$ ;

7 DEF Slice_selector( $s, d$ ):
8   INIT reservoir  $A[d] = [0 \dots d-1]$ ;
9   for  $i = d; i < s; i++$  do
10    Pseudorandom  $j$  out of  $[0 \dots i]$ ;
11    if  $j < d$  then
12       $A[j] = i$ ;
13   return index =  $A[d]$ ;
```

we use the **Jacobian matrix of the gradients** *w.r.t.* the input to reflect a sensitivity measure on the gradients. The input-gradient Jacobian is calculated by:

$$\mathbf{J}_l^G(X) = \frac{\partial \mathbf{g}_l(X)}{\partial X} = \frac{\partial}{\partial X} \left[\frac{\partial \ell(X, \mathbf{y}, \theta)}{\partial \theta_l} \right] \quad (1)$$

where $\mathbf{g}_l(\cdot)$ is to produce the gradients G_l of the l -th layer. $\ell(\cdot)$ is the loss over input X , ground truth \mathbf{y} , and parameters θ of the entire model, so $\mathbf{g}_l(\cdot)$ equals to the partial derivative of $\ell(\cdot)$, *w.r.t.* parameters θ_l in the l -th layer.

Then, given k data samples in the root dataset, we compute the Jacobian with **p -norm averaged over the data samples** as the information leakage risk in the l -th layer of gradients:

$$\mathcal{R}_l^X = \mathbb{E} \left[\left\| \mathbf{g}_l(X) - \mathbf{g}_l(X + \Delta X) \right\|_p \right] = \frac{1}{k} \sum_{i=1}^k \left\| \mathbf{J}_l^G(X_i) \right\|_p \quad (2)$$

where the choice of $\|\cdot\|_p$ reflects how to measure distance between two data samples. We use the F-norm since Jacobians are compared across layers with different sizes and F-norm will consider all dimensions of the data sample.

B. Risk Balancing based Random Perturbation

As mentioned above, information leakage risk from gradients can be layer-wisely determined in the initialization stage of learning. Note that the size of G_l still has an impact on the computed sensitivity. When reconstructing high-dimensional input data, **a large G_l size means more susceptible to attacks**.

Let S_l and \mathcal{R}_l be the size and leakage risk of the l -th layer. In order to balance the leakage risk weighted with layer size, we first use **the greatest common divisor** of S_l of each layer to partition the gradients into multiple blocks with the same size. Let S_b and \mathcal{R}_b be the size and leakage risk of the b -th block. Then, by solving a bin packing problem, the gradients can be **sliced as multiple slices**, each of which has balanced

leakage risk of original information. Note that the slicing can be determined in the initialization stage of learning.

Then, in each round of learning, each client will randomly select a certain proportion of slices from its shared gradients and add artificial noise to these slices. For gradients sliced into s slices, let $\gamma \in (0, 1]$ denote a perturbation ratio to determine the number of perturbed slices, equaling to $\max(1, \lfloor s \times \gamma \rfloor)$. The noise added to each randomly selected slice follows the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, in which the mean μ of noise variables is 0, the variance is σ^2 .

The overall procedure of local random perturbation is shown as Algorithm 1. Since the random perturbation is based on the *Reservoir Sampling*, correspondent computational complexity is bounded to $O(s)$, thereby should be lightweight. Therefore, with the easy-to-compute metric, sensitivity, we capture how information leaks in layer-wise manner, and design the perturbation with the weight of defense balanced across layers. Such that the quality of reconstructed data is severely degraded, while lightweight defense is maintained. Since all clients select their noise-adding slices locally and randomly, it is difficult for the attacker to locate these noisy slices, and then it is difficult to mute these slices and perform data reconstruction. Upon the proportion of noisy slices in the gradients exceeds a certain of level, even if the attacker may locate the noisy slices by brute force, the attacker will not be able to reconstruct the data.

V. GLOBAL UPDATE COMPENSATION

A. Attribute Correlation within Global Gradients

In *backward propagation*, the loss ℓ between the computed output and ground truth propagates from the last layer to the first layer. For l -th layer, the gradient vector \mathbf{G}_l consists of the gradients of the weights and biases which are computed using chain rule. Therefore, the layer-wise attributes in the global gradients are highly correlated because they are produced in backward propagation in layer-wise manner. We are thus able to derive, from the perturbed data, more accurate information about the original data as a compensation for the global update.

Since the attributes of the global gradients are highly correlated, these attributes have large variances in the directions of some vectors but small variances in the other directions. The addition of noise does not change the trends too much, because if the random numbers added to the data are independent, their variances will be evenly distributed among all the directions. The random noise added to the original data can be viewed as a random matrix and therefore its properties can be understood by studying the spectral properties of random matrices, i.e., the distribution of eigenvalues of the sample covariance matrix obtained from a random matrix [28].

Let \mathbf{C} denote the matrix of global gradients with perturbation, \mathbf{Z} denote the matrix of original data of \mathbf{C} , and \mathbf{R} denote the random matrix of noise variables. The matrices \mathbf{C} , \mathbf{Z} , and \mathbf{R} have the same shape of $v \times w$, and satisfy $\mathbf{C} = \mathbf{Z} + \mathbf{R}$. The elements of \mathbf{R} are *i.i.d.* random variables with the mean 0 and variance $\hat{\sigma}^2 = m \cdot \frac{d}{s} \sum_{i=1}^m \epsilon_i \sigma^2$, where m is the number of participants, s , d are the number of total slices and perturbed slices, respectively, σ^2 is the variance of noise, and $\epsilon \in (0, 1)$

is the aggregation weight. The covariance matrix of \mathbf{R} is given by $\mathbf{L}_R = \frac{1}{w} \mathbf{R}^T \mathbf{R}$. Clearly, \mathbf{L}_R is a $w \times w$ matrix.

Let $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_w$ be the eigenvalues of \mathbf{L}_R . Let the empirical cumulative distribution function of the eigenvalues $\lambda_i (1 \leq i \leq w)$ be denoted as following:

$$\Gamma(x) = \frac{1}{w} \sum_{i=1}^w \mathcal{U}(x - \lambda_i) \quad (3)$$

where $\mathcal{U}(x)$ is the unit step function, and $\mathcal{U}(x) = 1$ if $x > 0$, otherwise 0.

Then, in order to consider the asymptotic properties of the cumulative distribution function $\Gamma(x)$, we suppose the dimensions v , w of matrix \mathbf{R} to be functions $v(\cdot)$, $w(\cdot)$ of a variable ϖ . We then consider the asymptotics such that, in the limit as $\varpi \rightarrow \infty$, we have $v(\varpi) \rightarrow \infty$, $w(\varpi) \rightarrow \infty$ and $\frac{v(\varpi)}{w(\varpi)} \rightarrow Q$, where $Q \geq 1$. Under these assumptions, it can be shown that [28, 29] the empirical cumulative distribution function $\Gamma(x)$ converges in probability to a continuous distribution function $\Gamma_Q(x)$, the probability density function of which is given by:

$$\mathcal{F}_Q(x) = \frac{Q\sqrt{(x - \lambda_{\min})(\lambda_{\max} - x)}}{2\pi\hat{\sigma}^2 \cdot x}, x \in [\lambda_{\min}, \lambda_{\max}] \quad (4)$$

where $\mathcal{F}_Q(x) = 0$ if $x \notin [\lambda_{\min}, \lambda_{\max}]$, $\lambda_{\min} = (1 - 1/\sqrt{Q})^2 \cdot \hat{\sigma}^2$, and $\lambda_{\max} = (1 + 1/\sqrt{Q})^2 \cdot \hat{\sigma}^2$. The random matrix \mathbf{R} therefore has the probabilistic properties that can be used for the global update compensation.

B. Global Gradient Compensation

Given the matrix \mathbf{C} of the global gradients with perturbation, its covariance matrix \mathbf{L}_C satisfies:

$$\begin{aligned} v \cdot \mathbf{L}_C &= \mathbf{C}^T \mathbf{C} = \mathbf{P}_C \mathbf{\Lambda}_C \mathbf{P}_C^T = (\mathbf{Z} + \mathbf{R})^T (\mathbf{Z} + \mathbf{R}) \\ &= \mathbf{Z}^T \mathbf{Z} + \mathbf{R}^T \mathbf{Z} + \mathbf{Z}^T \mathbf{R} + \mathbf{R}^T \mathbf{R} \\ &= \mathbf{P}_Z \mathbf{\Lambda}_Z \mathbf{P}_Z^T + \mathbf{P}_R \mathbf{\Lambda}_R \mathbf{P}_R^T + \mathbf{R}^T \mathbf{Z} + \mathbf{Z}^T \mathbf{R} \end{aligned} \quad (5)$$

where \mathbf{P}_C , \mathbf{P}_Z , and \mathbf{P}_R are orthogonal matrices for which the column vectors are eigenvectors of $\mathbf{C}^T \mathbf{C}$, $\mathbf{Z}^T \mathbf{Z}$ and $\mathbf{R}^T \mathbf{R}$, and $\mathbf{\Lambda}_C$, $\mathbf{\Lambda}_Z$, and $\mathbf{\Lambda}_R$ are diagonal matrices with the corresponding eigenvalues on their diagonals. Since vectors of random matrix \mathbf{R} are generated by a statistically independent process and are uncorrelated with the vectors of global gradients, we have the expectations of $\mathbf{R}^T \mathbf{Z}$ and $\mathbf{Z}^T \mathbf{R}$ be equal to 0. If the number of samples is large enough, covariance matrix $\mathbf{P}_C \mathbf{\Lambda}_C \mathbf{P}_C^T$ can be simplified as $\mathbf{P}_Z \mathbf{\Lambda}_Z \mathbf{P}_Z^T + \mathbf{P}_R \mathbf{\Lambda}_R \mathbf{P}_R^T$. Then, based on results from the matrix perturbation theory as following:

Theorem 1. [30] Suppose that $\lambda_{1,a} \geq \lambda_{2,a} \geq \dots \geq \lambda_{w,a} \geq 0$ are the eigenvalues of $\mathbf{P}_C \mathbf{\Lambda}_C \mathbf{P}_C^T$, $\mathbf{P}_Z \mathbf{\Lambda}_Z \mathbf{P}_Z^T$ and $\mathbf{P}_R \mathbf{\Lambda}_R \mathbf{P}_R^T$, where $a \in \{c, z, r\}$. Then, for $1 \leq i \leq w$, $\lambda_{i,c} \in [\lambda_{i,z} + \lambda_{w,r}, \lambda_{i,z} + \lambda_{1,r}]$.

The above theorem provides a bound on the change in the eigenvalues of the data correlation matrix $\mathbf{P}_Z \mathbf{\Lambda}_Z \mathbf{P}_Z^T$ in terms of minimum and maximum eigenvalues of the noise correlation matrix $\mathbf{P}_R \mathbf{\Lambda}_R \mathbf{P}_R^T$. Suppose the data covariance matrix $\mathbf{P}_Z \mathbf{\Lambda}_Z \mathbf{P}_Z^T$ has only a few dominant eigenvalues, says, $\lambda_{1,z} \geq \dots \geq \lambda_{j,z}$, with $\lambda_{i,z} \leq \beta$ for some small value β , $j + 1 \leq i \leq w$. This condition is true for the global gradients. Suppose $\lambda_{j,z} > \lambda_{1,r}$,

the largest eigenvalue of the noise covariance matrix $P_R \Lambda_R P_R^T$. Then, we are able to approximately separate the data and noise eigenvalues Λ_Z, Λ_R from the eigenvalues Λ_C of the perturbed data by a simple threshold at $\lambda_{1,r}$.

The procedure of global gradient compensation as follows: First, **calculating the covariance matrix of the perturbed gradients C**. Using the distribution of eigenvalues of the covariance matrix according to the theory of random matrices, the covariance matrix of **C is decomposed into the noise and the data parts**. The eigenvalues corresponding to the data part are then used to compensate to actual global gradients. This can be done by simply projecting the data along the data eigenvectors and then mapping it back to the original space.

More specially, since the noise distribution has been given by a known variance $\hat{\sigma}^2$, Equation (4) can be used to calculate λ_{\min} and λ_{\max} , which provide the theoretical bounds of the **eigenvalues corresponding to noise**. Then, we calculate eigenvalues of covariance matrix $C^T C$, says, $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_w$. We identify the eigenvalues of the noise, $\lambda_i \geq \lambda_{i+1} \geq \dots \geq \lambda_j$, such that $\lambda_i \leq \lambda_{\max}$ and $\lambda_j \geq \lambda_{\min}$. The remaining eigenvalues are **eigenvalues corresponding to actual data**. Let $\Lambda_R^* = \text{diag}(\lambda_i, \lambda_{i+1}, \dots, \lambda_j)$ be the diagonal matrix with all the noise-related eigenvalues and P_R^* be the matrix the columns of which are correspondent eigenvectors. Similarly, let Λ_Z^* be the eigenvalue matrix for actual data and P_Z^* be correspondent eigenvector matrix. The compensation Z^* of the actual data Z is obtained by projecting C onto the subspace spanned by P_Z^* , i.e., $Z^* = C P_Z^* P_Z^{*T}$.

In order to understand how well the compensation works, let us consider the information loss when we select w^* dominant eigenvalues out of total w eigenvalues. All the variances along other $w - w^*$ directions will be lost. However, when the global gradients are highly correlated, the variances along the first w^* directions are much larger than variances along the rest $w - w^*$ directions. Therefore, removing those $w - w^*$ directions during the compensation does not cause much information loss. The information loss for the noise is different because the random numbers are independent for each attribute. Their correlations are 0. Therefore, their variances will be evenly distributed to those w^* directions. When we remove $w - w^*$ directions in the compensation, we are able to remove $\frac{w - w^*}{w}$ portion of noisy variances, i.e., migrating noise without much information loss.

C. Avoid Abusing Compensation

Considering that the server (i.e., the potential attacker) may abuse the compensation to upgrade the reconstructed results, i.e., the attacker may use compensation to **mitigate the noise of shared gradients** and then perform reconstruction. We target this concern with the *local clipping operation*, by **clipping the gradients** of individual data samples and **scaling them to the similar range corresponding to the noise**, and based on which to bound the influence of each individual data samples on the shared gradients G .

The procedure of local clipping operation as follows: First, suppose the added local noise follows the Gaussian distribution $\mathcal{N}(0, \sigma^2 \cdot B^2)$, where B is the noise intensity, determining the

range of noise. Then, at each step of local training, we use the *norm clip* suggested in [15] to clip the gradients of individual data samples according to their 2-norms, i.e., let the gradients g of every data sample be replaced by $g / \max(1, \frac{\|g\|_2}{B})$. This clipping ensures that if $\|g\|_2 \leq B$, then g is preserved, whereas if $\|g\|_2 > B$, it gets scaled down to be of the noise intensity B . At the server, it becomes hard for the clipped gradients even if being compensated to reveal whether a particular data sample has participated in the learning.

It is worth noting that the global compensation is still valid because the correlation information accumulated in the global gradients comes from multiple participants and is based on a number of data samples (thus redundant), which can clearly depict the strong correlation of the global gradients. Given that the noise variables have the unbiased estimate with the mean 0 and variance $\hat{\sigma}^2$. For simplicity, suppose that each client has the same aggregation weight, the global gradients will have an estimate of the mean with error scale $O(\hat{\sigma}/\sqrt{m})$, where m is the number of participants. As compared to the gradients shared from clients, the global gradients obtain a much higher signal-to-noise ratio, which facilitates global compensation.

D. Convergence Analysis

Let η denote the learning rate, and ϑ denote the accumulated errors and compensation in the global. Each client i calculates the local stochastic gradients $\Delta F(x_t; \xi_t^{(i)})$ based on the global model x_t and the local data samples $\xi_t^{(i)}$ in the iteration step t . Let $\Delta f(x_t)$ denote the gradients of global model x_t . In order to obtain the convergence rate, we first represent the updating rule for the global model x_t as following:

$$\begin{aligned} x_{t+1} - x_t &= -\eta \left[\frac{1}{m} \sum_{i=1}^m \Delta F(x_t; \xi_t^{(i)}) + \vartheta_{t-1} - \vartheta_t \right] \\ &= -\eta \Delta f(x_t) + \eta \zeta_t - \eta \vartheta_{t-1} + \eta \vartheta_t \end{aligned} \quad (6)$$

where $\zeta_t = \frac{1}{m} \sum_{i=1}^m [\Delta f(x_t) - \Delta F(x_t; \xi_t^{(i)})]$.

Assumption 1. 1) Suppose function $f(\cdot)$ is with L -Lipschitzian gradients, which means that $\|\Delta f(x) - \Delta f(y)\| \leq L\|x - y\|$, 2) Suppose the variance of the stochastic gradients is bounded, i.e., $\mathbb{E}\|\Delta F(x; \xi) - \Delta f(x)\|^2 \leq \phi^2$, 3) Suppose the magnitude of accumulated errors and global compensation is bounded by a constant ψ , i.e., $\mathbb{E}\|\vartheta_t\| \leq \frac{\psi}{2}$.

Note that the first and the second assumptions are commonly used in non-convex convergence analysis. The third assumption is used to restrict the scale of errors. Since we have from the assumptions $\mathbb{E}[\Delta F(x_t; \xi_t^{(i)})] = \Delta f(x_t)$, $\mathbb{E}\|\vartheta_t\|^2 \leq \psi^2$, and based on which it can be easily verified that for all t , we have $\mathbb{E}\zeta_t = 0$, and $\mathbb{E}\|\zeta_t\|^2 \leq \frac{\phi^2}{m}$.

Let the auxiliary sequence y_t be denoted as $y_t = x_t - \eta \vartheta_{t-1}$. The updating rule for y_t is given by $y_{t+1} - y_t = -\eta \Delta f(x_t) + \eta \zeta_t$. Since $f(\cdot)$ is with the L -Lipschitzian gradients, we have

$$\begin{aligned} \mathbb{E}\|\Delta f(y_t) - \Delta f(x_t)\|^2 &\leq L^2 \mathbb{E}\|y_t - x_t\|^2 \leq L^2 \eta^2 \psi^2 \quad (7) \\ \mathbb{E}f(y_{t+1}) - \mathbb{E}f(y_t) &\leq \mathbb{E}\langle y_{t+1} - y_t, \Delta f(y_t) \rangle + \frac{L}{2} \mathbb{E}\|y_{t+1} - y_t\|^2 = \end{aligned}$$

TABLE II
RESULTS OF MEASURE ON DEFENSES AGAINST DIFFERENT ATTACKS AND DIFFERENT DATASETS.

[A] Measure on Different Defenses against the DGA.												
	MNIST - ACC 91.69% without defenses				Fashion-MNIST - ACC 91.80% without defenses				CIFAR-10 - ACC 54.15% without defenses			
	Ours	GC	DP-G[-L]	PLD[-muted]	Ours	GC	DP-G[-L]	PLD[-muted]	Ours	GC	DP-G[-L]	PLD[-muted]
PSNR	9.41	9.52	9.36[9.39]	9.57[18.49]	9.66	9.83	9.57[9.62]	9.89[19.78]	9.61	9.79	9.55[9.52]	9.88[24.48]
SSIM	4.6E-2	5.1E-2	4.1E-2[4.3E-2]	5.3E-2[6.4E-1]	7.3E-2	7.7E-2	7.1E-2[6.5E-2]	8.2E-2[8.4E-1]	2.5E-2	2.6E-2	2.3E-2[2.4E-2]	2.9E-2[8.8E-1]
ACC	90.43%	36.52%	10.37%[10.21%]	87.77%[-]	89.29%	33.11%	10.10%[9.98%]	86.35%[-]	52.47%	29.84%	10.19%[10.00%]	49.91%[-]
ART	+8.45%	+4.63%	+3.91%[3.74%]	+14.52%[-]	+8.11%	+3.75%	+3.89%[4.04%]	+13.20%[-]	+8.97%	+3.58%	+4.03%[4.31%]	+14.09%[-]

[B] Measure on Different Defenses against the GIA.												
	MNIST - ACC 88.14% without defenses				Fashion-MNIST - ACC 86.57% without defenses				CIFAR-10 - ACC 49.31% without defenses			
	Ours	GC	DP-G[-L]	PLD[-muted]	Ours	GC	DP-G[-L]	PLD[-muted]	Ours	GC	DP-G[-L]	PLD[-muted]
PSNR	9.83	10.01	9.66[9.59]	10.43[19.61]	9.91	9.98	9.74[9.80]	10.14[21.23]	10.11	10.32	9.95[9.86]	10.79[27.04]
SSIM	4.9E-2	5.1E-2	4.4E-2[4.6E-2]	5.7E-2[7.3E-1]	7.5E-2	8.3E-2	6.8E-2[6.7E-2]	8.9E-2[9.5E-1]	4.1E-2	4.2E-2	3.0E-2[3.4E-2]	4.4E-2[9.3E-1]
ACC	86.87%	32.29%	10.46%[9.85%]	84.09%[-]	84.65%	30.38%	9.86%[9.77%]	81.10%[-]	47.73%	23.35%	10.01%[10.16%]	45.16%[-]
ART	+9.07%	+4.90%	+3.84%[3.66%]	+16.12%[-]	+8.62%	+4.23%	+4.14%[3.99%]	+15.86%[-]	+9.33%	+4.08%	+4.15%[4.02%]	+16.43%[-]

$$\begin{aligned}
& -\eta \mathbb{E} \langle \Delta f(\mathbf{x}_t), \Delta f(\mathbf{y}_t) \rangle + \eta \mathbb{E} \langle \zeta_t, \Delta f(\mathbf{y}_t) \rangle + \frac{L\eta^2}{2} \mathbb{E} \|\Delta f(\mathbf{x}_t) - \zeta_t\|^2 \\
& = -\eta \mathbb{E} \langle \Delta f(\mathbf{x}_t), \Delta f(\mathbf{y}_t) \rangle + \frac{L\eta^2}{2} \mathbb{E} \|\Delta f(\mathbf{x}_t)\|^2 + \frac{L\eta^2}{2} \mathbb{E} \|\zeta_t\|^2 \\
& \leq -\eta \mathbb{E} \langle \Delta f(\mathbf{x}_t), \Delta f(\mathbf{y}_t) \rangle + \frac{L\eta^2}{2} \mathbb{E} \|\Delta f(\mathbf{x}_t)\|^2 + \frac{L\eta^2 \phi^2}{2m} \quad (8) \\
& \leq \frac{-\eta + L\eta^2}{2} \mathbb{E} \|\Delta f(\mathbf{x}_t)\|^2 + 2L^2\eta^3\psi^2 + \frac{L\eta^2 \phi^2}{2m} \text{ by Equation (7)}
\end{aligned}$$

Let the learning rate η be $1/(2L + \phi\sqrt{T/m} + \psi^{2/3}T^{1/3})$. Then, by summing up the inequality above from $t = 0$ to $t = T - 1$, we have the following convergence rate under Assumption 1: $\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \|\Delta f(\mathbf{x}_t)\|^2 \leq \frac{\phi}{\sqrt{mT}} + \frac{1}{T} + \frac{\psi^{2/3}}{T^{2/3}}$, which factors constant.

VI. EXPERIMENTS

A. Experiment Setup

In our experiments, we evaluate proposed defensive mechanism against two different gradient leakage attacks under the non-IID settings, and show the results on three popular image datasets: MNIST, Fashion-MNIST, CIFAR-10.

Attack Methods. Gradient leakage attacks in [6] and [8] are used in our experiments. In the **deep gradient attack (DGA)** [6], the server optimizes reconstructed image to **minimize the Euclidean distance** between the real gradients and the dummy gradients that are generated by the reconstructed image in back propagation. The **gradient inverting attack (GIA)** [8] replaces the Euclidean distance function with the cosine similarity and performs the optimization on the sign of the gradients.

Defense Baselines. We compare proposed defensive mechanism with three existing defense methods: the **gradient compression (GC)** suggested in [6], the **differential privacy (DP)** [12], and the **most recent privacy leakage defense (PLD)** [18]. GC prunes gradients that are below a threshold magnitude such that a part of the gradients are preserved. DP protects privacy with theoretical bounds by adding noise to gradients. In our experiments, we separately apply the **Gaussian and Laplacian** noise as two DP baselines, *i.e.*, DP-G and DP-L. PLD provides with the provable defense by pruning a certain proportion of gradients of fully-connected layer.

Evaluation Metrics. We use the **peak signal-to-noise ratio (PSNR)** and **structural similarity index measure (SSIM)** between the reconstructed image and raw image to quantify the effectiveness of defenses. We use the **accuracy (ACC)** of

the global model on the testing set to measure the footprints of defenses. We use the **average round time (ART)** to check whether defenses are lightweight.

Datasets. We use three datasets: MNIST, Fashion-MNIST, CIFAR-10, which are popular on the image classification tasks. To simulate the settings of FL, we follow from [1] to distribute these datasets across clients in non-IID way. For MNIST and Fashion-MNIST, the datasets are distributed across 100 clients: each round of learning randomly selects 10 participants, each of which holds 2 random classes of data with 1~10 samples per class; the total of rounds is 200; local batch size is 1~20. For CIFAR-10, the dataset is distributed across 1000 clients: each round randomly selects 200 participants, each of which holds 2 random classes of data with 1~20 samples per class; the total of rounds is 2000; local batch size is 1~40. For other hyperparameters, we use SGD optimizer with learning rate of $1e-2$, and set the epoch to 2.

Models. We use the **LeNet model** suggested in [6], [8] and [18] for the DGA attack and the **ConvNet model** suggested in [18] for the GIA attack. The LeNet model is built with 4 convolutional layers followed by 1 fully-connected layer. The ConvNet model is built with 8 convolutional layers followed by 1 fully-connected layer. Note that the DGA and GIA have no demands on the model convergence status, *i.e.*, both two attacks can take place anytime during the training. Without loss of generality, we use randomly initialized models in our experiments. More details of experiment setup can be found in following subsections.

B. Experimental Results: from Main Perspective

For both two attacks, their **worst case** is that there is only one data sample in each batch and its label information has been known, where the quality of reconstructed images will be best upgraded. We evaluate all defenses in such a worst case to calculate correspondent PSNR and SSIM.

We choose **settings for different defensive methods**. Specially, for the GC, we set the pruning rate of gradients to 80% against the DGA and 90% against the GIA; for the DP, we set the noise variance to $1E-2$ both against the DGA and GIA; for the PLD, we set the pruning rate of the fully-connected layer to 60% against the DGA and 80% against the GIA; for our defense, we set the noise variance and perturbation rate to $1E-2$, 20% against the DGA and $1E-2$, 30% against the GIA.

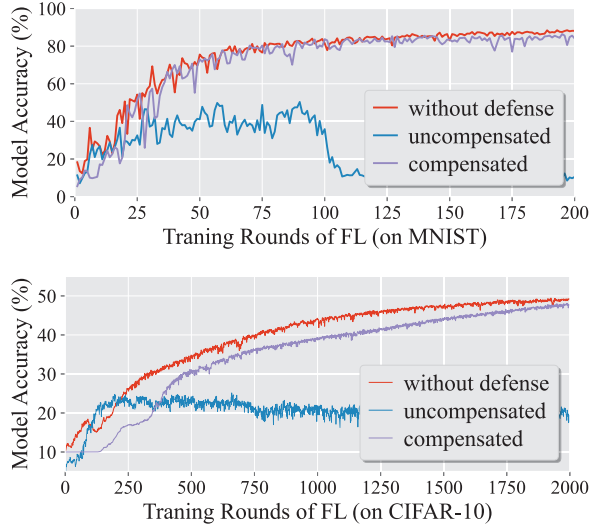


Fig. 2. Comparison between uncompensated and compensated defense.

We confirm above settings are the minimums for different defensive methods to provide the full defenses, where the raw images cannot be recognized from the reconstructed images. In addition, we apply *L-BFGS* optimizer with 500 iterations of reconstruction for the DGA, and apply *Adam* optimizer with 1000 iterations for the GIA.

We then evaluate convergence performance and overhead of training under the minimal defenses. As shown in Table II, our solution always achieves adequate defense with less than 2% accuracy loss and less than 10% overhead increment. PLD is also able to achieve a lower accuracy loss, however its rigid pattern of pruning, once being inferred and muted, will lead to significant increasing PSNR and SSIM which correspond to failed defense, where the raw images can be recognized from the reconstructed images. Compared with the muted PLD, our proposed defense can decrease the PSNR, SSIM between the reconstructed images and the raw images by as much as more than 60% for both two attacks. GC and DP can produce less overhead increment but their price of providing adequate protection is the complete loss of convergence performance.

It is worth noting that these experiments are based on the worst case of our defense (*i.e.*, there is only one sample in each batch). In other general cases (*i.e.*, more than one sample in each batch), our defense can achieve much better performance.

C. Power of Leakage Risk Quantifying

We investigate the importance of leakage risk quantification by using the the model and defense settings for the GIA. We use another baseline of slicing that is based on the layer size balancing instead of sensitivity balancing. In this baseline, we directly use the greatest common divisor of each layer size to partition the gradients into multiple slices with the same size. With the same perturbation ratio, in each round of training, the client randomly selects the same proportion of slices and adds the same Gaussian noise to these slices. We use a root dataset with 100 public data samples to quantify the gradient

TABLE III
RESULTS ON RISK QUANTIFICATION AND COMPENSATION.

[A] Gradient Sensitivity-based Slicing vs. Layer Size-based Slicing.						
MNIST		Fashion-MNIST		CIFAR-10		
	sensitivity	layer size	sensitivity	layer size	sensitivity	layer size
PSNR	9.83	14.51	9.91	13.70	10.11	15.02
SSIM	4.9E-2	1.2E-1	7.5E-2	2.6E-1	4.1E-2	1.5E-1
[B] Impact of Root Dataset <i>w.r.t.</i> its Size.						
MNIST		Fashion-MNIST		CIFAR-10		
	50	100	200	300	400	
PSNR	11.40	9.83	9.71	9.68	9.66	11.95
SSIM	6.2E-2	4.9E-2	4.5E-2	4.2E-2	4.0E-2	5.7E-2
						4.1E-2
[C] Impact of Root Dataset <i>w.r.t.</i> its Bias Probability.						
MNIST		Fashion-MNIST		CIFAR-10		
	0.1	0.2	0.4	0.6	1.0	
PSNR	9.89	10.32	10.53	11.68	13.66	10.41
SSIM	5.0E-2	5.4E-2	5.7E-2	6.9E-2	9.1E-2	4.3E-2
						4.5E-2
[D] Impact of Q on Convergence Performance.						
MNIST		Fashion-MNIST		CIFAR-10		
	1	6	12	20	1	6
ACC	82.92%	86.87%	86.99%	86.10%	41.64%	47.73%
						47.85%
[E] Local Perturbation with Clipping vs. Local Perturbation without Clipping.						
MNIST		Fashion-MNIST		CIFAR-10		
	clipping	no clipping	clipping	no clipping	clipping	no clipping
PSNR	9.95	12.36	10.02	12.74	10.66	13.28
SSIM	5.2E-2	9.9E-2	7.7E-2	9.5E-2	4.8E-2	8.7E-2
ACC	86.03%	86.87%	83.87%	84.65%	47.15%	47.73%
ART	+1.32%	-	+0.98%	-	+1.14%	-

sensitivity, *i.e.*, the information leakage risk. From the results shown in Table III[A], we find that the gradient sensitivity based slicing can capture more accurately the risk of original information leakage from gradients, thereby the correspondent PSNR, SSIM are significantly lower than the baseline.

We then study the impact of the root dataset on quantifying leakage risk *w.r.t.* its size and how it is sampled. Table III[B] shows PSNR, SSIM when the size of the root dataset increases from 50 to 400, where the root dataset is sampled uniformly in original dataset. We observe that a root dataset with only 100 training samples is sufficient to capture the risk of information leakage. When the size of the root dataset increases beyond 100, correspondent effectiveness of defense further increases slightly. We also evaluate the impact of the root dataset *w.r.t.* its bias probability. Table III[C] shows PSNR, SSIM when the bias probability varies. We increase the bias probability from 0.1 to 1.0 to simulate the difference between the root dataset distribution and the global dataset distribution. We observe that the quantification of leakage risk is accurate and robust when the bias probability is not too large. Therefore, the leakage risk quantification can work well when the root dataset distribution does not diverge too much from the global data distribution.

D. Insights on Compensation

We also evaluate the effect of compensation on the MNIST and CIFAR-10 datasets. We use the model and defense settings for the GIA. From the results shown in Figure 2, we find that the compensated cases have achieved the similar accuracy to the cases without defense, while the uncompensated cases have completely lost the accuracy. When doing compensation, we first flatten the global gradients into a single-column vector, splitting it into a fixed number of vectors with equal length according to Q in Equation (4), and we append these vectors to form a matrix. The compensation is then applied to this

matrix. We evaluate the impact of Q on the effectiveness of compensation. Table III[D] shows the model accuracy when the value of Q increases from 1 to 20. We observe that Q being 6 ~ 12 is sufficient to characterize the correlation of the global gradients, in which the compensation errors reduce to relatively stable values. When the value of Q increases beyond 6, correspondent convergence performance further increases slightly and even drops for a larger Q .

Considering that the attacker might abuse compensation to upgrade the reconstructed results, *i.e.*, the attacker might use the compensation to mitigate the noise of shared gradients and then perform data reconstruction. Clipping these gradients in the local training can target this concern. We investigate its effect by using the model and defense settings for the GIA. Then, we show correspondent results in Table III[E]. From the results, we find that the clipping can protect the compensation from being abused. With gradient clipping, the raw images will not be recognized from reconstructed images, corresponding to a lower PSNR, SSIM. Notably, the accuracy loss caused by clipping is less than 1%, and correspondent overhead increment is less than 1.5%.

VII. RELATED WORK

A. Gradient Leakage Attack

Reconstruction of image data from gradient information was first discussed in [10] for the neural networks, who prove that reconstruction is possible for a single neuron or linear layer. Based on this, Wang *et al.* [31] show that reconstruction of a single image is possible for a 4-layered CNN consisting of a significantly large fully-connected layer. Zhu *et al.* [6] extend this to reconstruct training samples from a similar 4-layer CNN, and show that the label information can also be jointly reconstructed. Following up [6], Zhao *et al.* [7] note that label information can be inferred analytically from the gradients of the last one layer, thereby makes reconstruction easier. Geiping *et al.* [8] move towards the reconstruction of multiple images from their averaged gradients. They replace the Euclidean distance function with the cosine similarity and perform the optimization on the sign of the gradients. Yin *et al.* [19] further explore how to reconstruct individual images in a batch, given averaged gradients. They present a framework, using training images from a larger batch (up to 48 images) for a large model *e.g.*, ResNet50, on a complex dataset *e.g.*, ImageNet (with 1000 class labels, resolution of 224×224).

B. Protection on Data Privacy

A potential solution for protecting data privacy is to leverage cryptographic approaches, *e.g.*, secure multiparty computation (MPC), homomorphic encryption, or differential privacy. MPC based techniques mainly utilize the garbled circuits or secret sharing, allowing multiple parties to collaboratively compute a function in a protocol that each party knows nothing except its input and output (*i.e.*, meeting the zero-knowledge definition) [16]. Their critical bottleneck is the high communication cost, and communication-efficient implementations require either extensive offline computation or lower the privacy protection.

Bonawitz *et al.* [17] present a secure aggregation and ensure that the server can only learn the aggregated data instead of the individual data from any clients. However, in each rounding of learning, clients must synchronize secret keys and zero-sum masks, imposing a strong need on synchronous training.

Homomorphic encryption [9–11] use a cryptographic computation method that can enable the certain computation (*e.g.*, addition) to be performed directly on encrypted data without decrypting them first. This protects the computed data from being exposed to any external parties including the server as computation is performed on ciphertexts. However, the privacy of homomorphic encryption relies on the size of the encrypted data (*i.e.*, more privacy requires a larger encrypted data size), and performing the computations in the large encrypted data is computationally inefficient.

Differential privacy [12–15] use a noisy release mechanism that enables clients to perform a differentially private transformation, and ensure that each individual data sample from local datasets hard to be identified, by injecting artificial noise at the client side. This entails tradeoff between convergence performance and privacy protection, however it means that stronger privacy protection leads to significant training accuracy loss.

Different from general-purpose protection as above, Sun *et al.* [18] focus on defending privacy against a particular attack domain *i.e.*, the gradient leakage attack in FL. They provide with a provable defense by pruning a certain proportion of gradients of fully-connected layer. Although this defense seems valid under such an assumption: the attacker will naively use entire gradients for reconstruction. However, the defense was found in vain once the attacker mutes the certain layer, while the remaining gradients without perturbation still expose the privacy to the attacker. Thus in practice, it fails short in terms of privacy protection.

VIII. CONCLUSION

In this paper, we propose a brand new defensive mechanism against gradient leakage attacks in FL, simultaneously aiming at the following three key criteria: 1) *lightweight* in terms of defense overhead; 2) *guaranteed* in terms of training accuracy; and 3) *adequate* in terms of privacy protection. Overall seen from experimental results, without sacrificing training accuracy, our lightweight defense can decrease the PSNR, SSIM between the reconstructed images and raw images by as much as more than 60% for two gradient leakage attacks, compared with baseline defensive methods.

ACKNOWLEDGE

This work was supported by National Key Research and Development Program of China under Grant 2019YFB2102404, Hong Kong RGC Research Impact Fund (RIF) with the Project No. R5060-19, General Research Fund (GRF) with the Project No. 152221/19E, 152203/20E, and 152244/21E, the National Natural Science Foundation of China under Grant 61872310, 62072069, and 61772112, Shenzhen Science and Technology Innovation Commission (R2020A045).

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 *Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017, pp. 1273–1282.
- [2] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.
- [3] F. Granqvist, M. Seigel, R. van Dalen, Á. Cahill, S. Shum, and M. Paulik, "Improving on-device speaker verification using federated learning with privacy," *arXiv preprint arXiv:2008.02651*, 2020.
- [4] S. Ramaswamy, R. Mathews, K. Rao, and F. Beaufays, "Federated learning for emoji prediction in a mobile keyboard," *arXiv preprint arXiv:1906.04329*, 2019.
- [5] W. Wei, L. Liu, M. Loper, K.-H. Chow, M. E. Gursoy, S. Truex, and Y. Wu, "A framework for evaluating client privacy leakages in federated learning," in *Proceedings of European Symposium on Research in Computer Security (ESORICS)*, 2020, pp. 545–566.
- [6] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, pp. 14 774–14 784, 2019.
- [7] B. Zhao, K. R. Mopuri, and H. Bilen, "idlg: Improved deep leakage from gradients," *arXiv preprint arXiv:2001.02610*, 2020.
- [8] 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 (NeurIPS)*, pp. 16 937–16 947, 2020.
- [9] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning," in *Proceedings of USENIX Annual Technical Conference (ATC)*, 2020, pp. 493–506.
- [10] 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 (TIFS)*, vol. 13, no. 5, pp. 1333–1345, 2017.
- [11] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, D. Papadopoulos, and Q. Yang, "Secureboost: A lossless federated learning framework," *IEEE Intelligent Systems*, 2021.
- [12] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.
- [13] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.
- [14] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 15, pp. 3454–3469, 2020.
- [15] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2016, pp. 308–318.
- [16] P. Mohassel and Y. Zhang, "Secureml: A system for scalable privacy-preserving machine learning," in *Proceedings of IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 19–38.
- [17] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *proceedings of ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2017, pp. 1175–1191.
- [18] J. Sun, A. Li, B. Wang, H. Yang, H. Li, and Y. Chen, "Provable defense against privacy leakage in federated learning from representation perspective," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 9311–9319.
- [19] 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 16 337–16 346.
- [20] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.
- [21] K. Leino and M. Fredrikson, "Stolen memories: Leveraging model memorization for calibrated white-box membership inference," in *Proceedings of USENIX Security Symposium (Security)*, 2020, pp. 1605–1622.
- [22] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song, "The secret sharer: Evaluating and testing unintended memorization in neural networks," in *Proceedings of USENIX Security Symposium (Security)*, 2019, pp. 267–284.
- [23] B. Neyshabur, S. Bhojanapalli, D. Mcallester, and N. Srebro, "Exploring generalization in deep learning," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, pp. 5947–5956, 2017.
- [24] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, "Certified robustness to adversarial examples with differential privacy," in *Proceedings of IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 656–672.
- [25] R. Novak, Y. Bahri, D. A. Abolafia, J. Pennington, and J. Sohl-Dickstein, "Sensitivity and generalization in neural networks: an empirical study," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.
- [26] J. Sokolić, R. Gyires, G. Sapiro, and M. R. Rodrigues, "Robust large margin deep neural networks," *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp. 4265–4280, 2017.
- [27] T. Wang, J.-Y. Zhu, A. Torralba, and A. A. Efros, "Dataset distillation," *arXiv preprint arXiv:1811.10959*, 2018.
- [28] J. Yao, S. Zheng, and Z. Bai, *Sample covariance matrices and high-dimensional data analysis*. Cambridge University Press Cambridge, 2015.
- [29] D. Jonsson, "Some limit theorems for the eigenvalues of a sample covariance matrix," *Journal of Multivariate Analysis*, vol. 12, no. 1, pp. 1–38, 1982.
- [30] H. Weyl, "Inequalities between the two kinds of eigenvalues of a linear transformation," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 35, no. 7, p. 408, 1949.
- [31] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, 2019, pp. 2512–2520.