PPIDSG: A Privacy-Preserving Image Distribution Sharing Scheme with GAN in Federated Learning

Yuting Ma[1], **Yuanzhi Yao**[2*], **Xiaohua Xu**[1*]

[1]University of Science and Technology of China
[2]Hefei University of Technology
ytma@mail.ustc.edu.cn, yaoyz@hfut.edu.cn, xiaohuaxu@ustc.edu.cn

Abstract

Federated learning (FL) has attracted growing attention since it allows for privacy-preserving collaborative training on decentralized clients without explicitly uploading sensitive data to the central server. However, recent works have revealed that it still has the risk of exposing private data to adversaries. In this paper, we conduct reconstruction attacks and enhance inference attacks on various datasets to better understand that sharing trained classification model parameters to a central server is the main problem of privacy leakage in FL. To tackle this problem, a privacy-preserving image distribution sharing scheme with GAN (PPIDSG) is proposed, which consists of a block scrambling-based encryption algorithm, an image distribution sharing method, and local classification training. Specifically, our method can capture the distribution of a target image domain which is transformed by the block encryption algorithm, and upload generator parameters to avoid classifier sharing with negligible influence on model performance. Furthermore, we apply a feature extractor to motivate model utility and train it separately from the classifier. The extensive experimental results and security analyses demonstrate the superiority of our proposed scheme compared to other state-of-the-art defense methods. The code is available at https://github.com/ytingma/PPIDSG.

Introduction

Federated learning (McMahan et al. 2017), which enables clients to train their data locally and upload only model parameters to the server for model aggregation, undoubtedly plays a significant role in autonomous driving (Xie et al. 2022), health care (Guo et al. 2021; Liu et al. 2021; Jiang, Wang, and Dou 2022), and other industries in recent years. However, recent researches (Geng et al. 2022; Fu et al. 2022; Yu et al. 2023; Zhu, Yao, and Blaschko 2023) demonstrate that adversaries can utilize the parameter information uploaded by clients to carry out attacks, resulting in serious privacy leakage problems.

Several strategies have been proposed to enhance security in the face of privacy threats, such as homomorphic encryption (Phong et al. 2018; Jin et al. 2023) can use an encryption algorithm that satisfies the homomorphic operation property of ciphertext to encrypt shared model parameters and differential privacy (Abadi et al. 2016; Liao et al. 2023) can prevent privacy leakage by introducing random noise. Gradient perturbation (Sun et al. 2021) aims to perturb data representation to guarantee privacy. However, the aforementioned methods of protecting sensitive data in FL significantly increase computational overhead or sacrifice efficiency to safeguard sensitive data. Another possible way is to encrypt training data (Chuman, Sirichotedumrong, and Kiya 2019; Huang et al. 2020) without greatly affecting model accuracy. These solutions mentioned above all depend on sharing classifier parameters to fulfill the model aggregation against image reconstruction attacks in federated learning. However, in addition to image reconstruction attacks, clients are vulnerable to membership inference attacks and label inference attacks. These defenses ignore inference attacks (Shokri et al. 2017; Geng et al. 2022) in federated learning and cannot provide adequate security.

To further show the problem of privacy-preserving methods in FL, we first begin with a theoretical analysis of label inference attacks and then apply these defenses to perform attacks on a variety of datasets. Because FL mitigates overfitting after model aggregation, we next propose an upgraded membership inference attack that integrates two shadow models instead of a single model considering the server attacker can access model parameters as well as structures. We also experimentally investigate the feasibility of image reconstruction attacks in FL after protection by defense methods. Extensive experiments and theoretical studies reveal valuable observations into the relationship between privacy leakage and trained classification model parameter sharing.

Motivated by these observations, we design PPIDSG, a scheme that can defend against privacy leakage while maintaining model utility. Specifically, 1) *How to realize privacy protection while simultaneously resisting attacks?* We leverage a block scrambling-based encryption algorithm to transform the original image domain into the target domain while training the classifier locally; 2) *How to enable federated learning without uploading the classifier parameters?* We utilize a GAN to capture the target image distribution and upload the generator parameters to share it; 3) *How to maintain classification accuracy?* We add an additional classification loss to our generator and introduce an independently trained autoencoder to extract interesting features. To summarize, this paper makes the following contributions:

[*]Corresponding authors.

- We present an enhanced membership inference attack in FL by reconstructing classifiers of a victim and others through uploaded model parameters.
- Theoretical analyses and practical experiments validate that parameter sharing of trained classifiers leads to privacy leakage in federated learning.
- To the best of our knowledge, a framework combining a GAN and a feature extractor without uploading a trained classifier is first proposed, achieving the balance between privacy-preserving and model utility in FL.
- Extensive experiments on four datasets compare our scheme with other defenses and results manifest our approach provides a considerably more secure guarantee without compromising accuracy.

Related Work

GAN in Privacy Applications

GAN (Goodfellow et al. 2014) was first proposed in 2014. The adversarial loss between a discriminator and a generator, which attempts to generate images that are indistinguishable from the real, is GAN's successful secret. CycleGAN (Zhu et al. 2017) proposed a cycle consistency loss to complete the image-to-image translation with unpaired images. Owing to GAN's advantage in visual translation, more researchers are considering employing it for defense or attack. A real-time GAN-based learning procedure (Hitaj, Ateniese, and Perez-Cruz 2017) was proposed that enables the adversary to generate samples from the target distribution. DaST (Zhou et al. 2020) utilized GAN to train a substitute model and launched a model extraction attack. DeepEDN (Ding et al. 2021) suggested a medical image encryption and decryption network based on CycleGAN. FedCG (Wu et al. 2022) leveraged a conditional GAN to achieve privacy protection against image reconstruction attacks.

Attack in Federated Learning

Traditional attacks in FL include membership inference attacks, property inference attacks, and image reconstruction attacks. MIA (Shokri et al. 2017) raised the membership inference attack: given the black-box access to a model, determine whether a given data record is included in the target dataset. To build an inference attack model, the adversary needs to create shadow training data for shadow models. ML-Leaks (Salem et al. 2019) relaxed the assumption and extended it to more scenarios. Later, additional technologies (Hayes et al. 2019; Melis et al. 2019; Duan et al. 2023) were employed to raise the attack, such as GAN.

Property inference attacks infer particular attributes that hold only for a subset of training data and not for others. The attribute can be replaced with labels (Fu et al. 2022). Image reconstruction attacks exploit gradients that users submitted to the server to restore original samples. With the guidance of the gradient difference produced by original images and dummy images, DLG (Zhu, Liu, and Han 2019) carried out the minimization optimization. The extraction of ground-truth labels is first proposed in iDLG (Zhao, Mopuri, and Bilen 2020) as an approach to strengthen the attack. In GradInversion (Yin et al. 2021), a label recovery algorithm for data in larger batches and a group consistency regularization were utilized to rebuild complex images with high fidelity. Then, a zero-shot approach (Geng et al. 2022) promoted image reconstruction to distributed learning and restored labels even when a batch of labels has duplicate labels.

Privacy Leakage in FL

Attack Setup

In this paper, we consider a common attack scenario where the attacker is an honest-but-curious server. It indicates that the attacker adheres to the federated learning protocol without corrupting the training process. Clients upload their local parameters (gradients or weights) to the server. These two parameters can be regarded as comparable when all users train only one local epoch between two global aggregations with their all training data. In this assumption, the attack process is equivalent to a white-box attack, where the attacker acquires the knowledge of model parameters and structures.

Label Inference Attack

Each client C_k has a local dataset $\mathcal{D}_k = \{(x_i, y_i)\}_{i=1}^{n_k}$, where each sample (x_i, y_i) has a data sample x_i and a ground truth label y_i, and they select *bs* (batch size) of their local datasets for training. Since most classifiers categorize via the cross-entropy loss function, we define the gradients of loss function w.r.t. network weights $\mathcal{W}$ as:

$$\nabla_{\mathcal{W}}\mathcal{L}(\mathbf{x},\mathbf{y}) = -\frac{1}{bs}\sum_{i=1}^{bs}\sum_{j=1}^{n_c}\nabla_{\mathcal{W}}[y_i(j)\log y_i'(j)], \qquad (1)$$

where y_i' is the logit output of the last layer after softmax and n_c is the number of label categories. When the index j of output is equal to the ground truth, $y_i(j) = 1$, else $y_i(j) = 0$. Thus, our goal is to measure the number of images $\sum_{i=1}^{bs} y_i(j)$ in each label category j. For example, if we observe that $\sum_{i=1}^{bs} y_i(0) = 2$ and $\sum_{i=1}^{bs} y_i(1) = 2$ in the MNIST dataset which has a batch size of four, then we conjecture that there are two labels "0" and two labels "1".

According to GradInversion (Yin et al. 2021), the gradient of each x_i w.r.t. the network output z_i at index j is $\nabla_{z_i(j)}\mathcal{L}(x_i, y_i) = y_i'(j) - y_i(j)$. To achieve our goal from uploading gradients, we define $\mathcal{W}_{m,j}$ as the weights of m^{th} unit of the last hidden layer to the output layer at j index and thus obtain the following equation by using a chain rule:

$$\begin{aligned}\sum_{m=1}^{n_m}\nabla_{\mathcal{W}_{m,j}}\mathcal{L}(\mathbf{x},\mathbf{y}) &= \frac{1}{bs}\sum_{i=1}^{bs}\sum_{m=1}^{n_m}\frac{\partial\mathcal{L}(x_i,y_i)}{\partial z_i(j)}\cdot\frac{\partial z_i(j)}{\partial\mathcal{W}_{m,j}}\\ &= \frac{1}{bs}\sum_{i=1}^{bs}(y_i'(j) - y_i(j))\sum_{m=1}^{n_m}\frac{\partial z_i(j)}{\partial\mathcal{W}_{m,j}},\end{aligned} \qquad (2)$$

where $\frac{\partial z_i(j)}{\partial\mathcal{W}_{\mathrm{m,j}}} = o_{m,i}$ is m^{th} input of the fully-connected layer with an image x_i and n_m is the dimension number of the last hidden layer. However, the above equation has two unknown values $y_i'(j)$ and $o_{m,i}$ in addition to the desired one $y_i(j)$. We estimate the unknown values by feeding random samples to the classification model multiple times and derive sample numbers in each category j:

$$\sum_{i=1}^{bs} y_i(j) \approx f(\sum_{i=1}^{bs} \tilde{y}'_i(j) - \frac{bs \cdot \sum_{m=1}^{n_m} \nabla_{\mathcal{W}_{\mathrm{m,j}}} \mathcal{L}(\mathbf{x}, \mathbf{y})}{\tilde{o}_i}), \quad (3)$$

where $\tilde{o}_i = \sum_{m=1}^{n_m} mean(\tilde{o}_{m,i})$, $\tilde{y}'_i$ and $\tilde{o}_{m,i}$ are generated from random samples, and $f(\cdot)$ is a mode function from multiple epochs to improve attacks. Attackers obtain the gradient by using the uploaded classifier parameters, which can then be utilized to launch attacks and cause privacy leakage.

Membership Inference Attack

The adversary C_A possesses a dataset $\mathcal{D}_{shadow}$ that includes some data records from the same distribution that are not in the target dataset. This security assumption is strong, yet it maximizes detecting the protection of defenses. The attack relies on overfitting caused by the trained classification model, which can be acquired from the submitted parameters. Considering that overfitting has been mitigated by model aggregation in FL, we improve the attack by rebuilding different shadow models of a victim and other users.

Given these uploaded model parameters of clients, C_A produces copies of the victim model and other models, which we denote as M^{victim} and M^{others}. If client number $K > 2$, C_A aggregates all models of other users as M^{others}. As depicted in Figure 1, C_A randomly divides the $\mathcal{D}_{shadow}$ into two disjoint sets: $\mathcal{D}^{victim}_{shadow}$ and $\mathcal{D}^{others}_{shadow}$, and then inputs them into M^{victim} and M^{others} respectively, generating the prediction vectors p^{victim}_{shadow} and p^{others}_{shadow}. The former is manually labeled in (member) and the latter is labeled out (non-member). Next, C_A trains the inference model M_{attack} with (p^{victim}_{shadow}, in) and (p^{others}_{shadow}, out). The attacker has a skeptical dataset and is unable to distinguish between data coming from the victim and other users. Finally, the adversary feeds the skeptical dataset into trained M_{attack}, and the success rate is the percentage of correctly inferred data records.

Image Reconstruction Attack

Several image recovery optimization functions minimize the gradient difference between the victim uploaded gradient $\nabla\mathcal{W}$ and the generated gradient by dummy images x^* with dummy or inferred labels y^*:

$$x^* = \arg\min_x \left\| \frac{\partial \mathcal{L}(x, y^*; \mathcal{W})}{\partial \mathcal{W}} - \nabla\mathcal{W} \right\|^2. \quad (4)$$

We conduct a theoretical investigation of these attacks described above, which are caused by sharing trained classifier parameters and result in privacy leakage. In the Experiments section, we report detailed comparison attack results under the protection of our proposal and other defense methods.

Methodology

In PPIDSG, we aim to achieve three objectives: 1) Privacy Objective: safeguard users' privacy and resist attacks; 2) Parameter Objective: avoid trained classifier parameters from being uploaded when finishing global training in FL; 3) Utility Objective: maintain the model's classification accuracy.

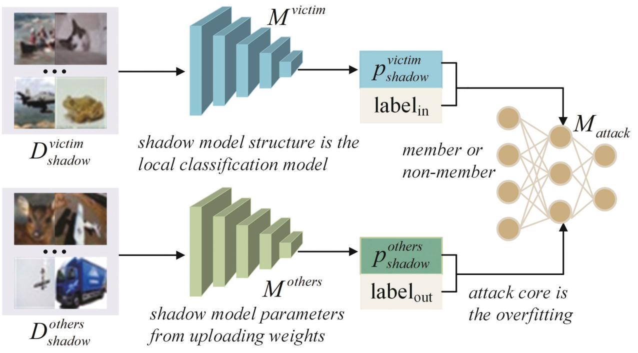

Figure 1: The enhanced membership inference attack.

Framework of PPIDSG

Figure 2 illustrates our privacy-preserving framework which mainly consists of three modules: 1) a block scrambling-based encryption algorithm; 2) an image distribution sharing method; 3) local classification training. To satisfy "Privacy Objective", we employ the encryption algorithm to convert original images into encrypted images (target domain) and train the classifier (C) locally. Additionally, data augmentation (Cubuk et al. 2019) expands original samples while protecting privacy more effectively. To fulfill "Parameter Objective", we deploy the distribution sharing method based on CycleGAN by transmitting only the parameters of a generator (G) instead of C to the server. We train G with a discriminator (D) to capture image distribution and eliminate the cycle consistency loss to prevent G from reconstructing target images into original images. To achieve "Utility Objective", we introduce a feature extractor (F) that utilizes an autoencoder (Sellami and Tabbone 2022) to extract interesting features and train it separately from C. Then, a classification loss is applied to G to focus on the distribution over image categories, for a better classification utility.

We consider a federated system with K clients. Each client C_k has a local dataset $\mathcal{D}_k$. There are N samples in total. Each client in our paper trains its models locally, uploading only the generator parameters W to the central server. Then, the server simply aggregates the model parameters in $(t+1)^{th}$ communication epoch by:

$$W^{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{N} (W^t - \alpha \nabla \mathcal{L}(W^t; \mathcal{D}_k)), \quad (5)$$

where α and $\mathcal{L}$ represent the learning rate and loss function.

Block Scrambling-based Encryption

In this paper, we upload parameters of G that may be utilized by an attacker to generate images that are similar to the target domain to execute attacks, thus we need to encrypt the target domain before training. According to DeepEDN (Ding et al. 2021), an image can be regarded as encrypted if it can be transformed into a domain that is quite dissimilar to the original. While we can apply any transformation methods to encrypt images, we prefer the block scrambling-

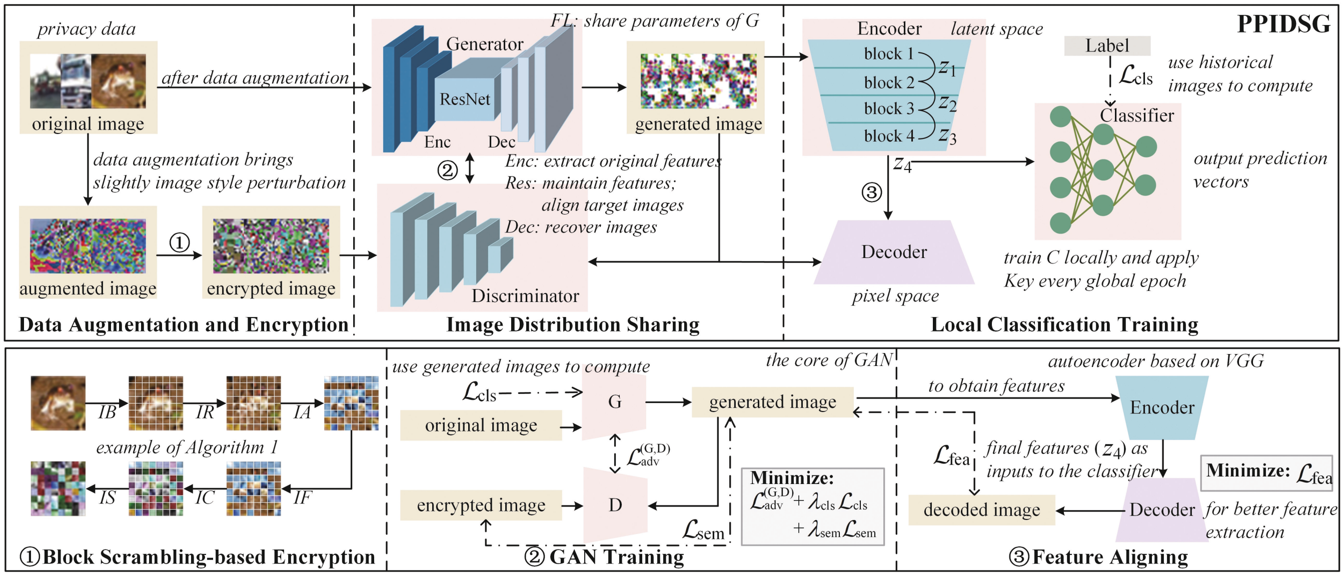
privacy data
original image
after data augmentation
data augmentation brings slightly image style perturbation
augmented image
①
encrypted image
Data Augmentation and Encryption
FL: share parameters of G
Generator
ResNet
Enc
Dec
generated image
②
Enc: extract original features
Res: maintain features; align target images
Dec: recover images
Discriminator
Image Distribution Sharing
Encoder
latent space
block 1
block 2
block 3
block 4
z_1
z_2
z_3
z_4
③
Decoder
pixel space
PPIDSG
Label
$\mathcal{L}_{cls}$
use historical images to compute
Classifier
output prediction vectors
train C locally and apply Key every global epoch
Local Classification Training
IB
IR
IA
example of Algorithm 1
IS
IC
IF
①Block Scrambling-based Encryption
use generated images to compute
$\mathcal{L}_{cls}$
G
original image
$\mathcal{L}_{adv}^{(G,D)}$
encrypted image
D
$\mathcal{L}_{sem}$
the core of GAN
generated image
Minimize: $\mathcal{L}_{adv}^{(G,D)} + \lambda_{cls}\mathcal{L}_{cls} + \lambda_{sem}\mathcal{L}_{sem}$
②GAN Training
autoencoder based on VGG
to obtain features
Encoder
$\mathcal{L}_{fea}$
final features (z_4) as inputs to the classifier
Minimize: $\mathcal{L}_{fea}$
decoded image
Decoder
for better feature extraction
③Feature Aligning

Figure 2: The overview of our proposed framework.

based encryption algorithm taking into account the model utility and encryption timeliness:

- IR: The resulting block can be rotated 0, 90, 180, and 270 degrees.
- IA: The L-bit pixel $p(i)$ can be adjusted with a pseudo-random bit r_i. The new pixel $\hat{p}(i)$ is calculated by

$$\hat{p}(i) = \begin{cases} p(i), & r_i = 1 \\ p(i) \oplus (2^L - 1), & r_i = 0 \end{cases} . \tag{6}$$

- IF: The resulting block can be flipped horizontally or vertically or not.
- IC: The colored image block can exchange pixel values in three color channels.

As illustrated in Algorithm 1, the above operations are controlled by random keys. We denote the distribution of the original image domain X as $x_i \sim p_{data}(x)$ and the distribution of the encrypted image domain $\hat{X}$ as $\hat{x}_i \sim p_{data}(\hat{x})$. Encrypted images are regarded as a target domain.

Algorithm 1: Block Scrambling-based Encryption.

Input: original image x_i, randomly key $K_j, j \in \{1, \cdots, 5\}$.
Output: encrypted image $\hat{x}_i$.

```
IB($x_i$): $x_i$ with $P_x \times P_y$ pixels be divided into $n$ non-overlapped blocks $\mathcal{B}_l^{(0)}$ with $B_x \times B_y$ pixels, $l \in [1, n]$.
for each $\mathcal{B}_l^{(0)}$, $l \in [1, n]$ do
    $\mathcal{B}_l^{(1)} \leftarrow$ IR($\mathcal{B}_l^{(0)}$, $K_1$);
    $\mathcal{B}_l^{(2)} \leftarrow$ IA($\mathcal{B}_l^{(1)}$, $K_2$);
    $\mathcal{B}_l^{(3)} \leftarrow$ IF($\mathcal{B}_l^{(2)}$, $K_3$);
    $\mathcal{B}_l^{(4)} \leftarrow$ IC($\mathcal{B}_l^{(3)}$, $K_4$); //optional
end for
IS($\mathcal{B}_l^{(4)}$, $K_5$): Shuffle and assemble all blocks $\mathcal{B}_l^{(4)}$ to generate a new encrypted image $\hat{x}_i$ with $K_5$.
return $\hat{x}_i$
```

Image Distribution Sharing

The fundamental idea of CycleGAN is to transfer original images into target images while capturing the target distribution. Inspired by the idea, we utilize the structure to perform image distribution capture and upload parameters to finish FL. Also, using the original image instead of noise as the original domain can further enhance privacy protection. G consists of three components: an encoder, ResNet blocks, and a decoder. The encoder, consisting of convolutional layers, is employed to extract features from original images. ResNet blocks are mainly responsible for maintaining original features and aligning them toward target images. The decoder consists of deconvolution and convolutional layers, which restores the feature vector to the image. By distinguishing generated images from target images, D aims to improve the image distribution capture performance. We apply an adversarial loss without conditional labels to train G and D. Given training samples $\{x_i\}_{i=1}^{bs}$ and $\{\hat{x}_i\}_{i=1}^{bs}$, the objective can be written as:

$$\begin{aligned} \mathcal{L}_{adv}^{(G,D)} &= E_{\hat{x}_i \sim p_{data}(\hat{x})} \left[\log D(\{\hat{x}_i\}_{i=1}^{bs}) \right] \\ &+ E_{x_i \sim p_{data}(x)} \left[\log(1 - D(G(\{x_i\}_{i=1}^{bs}))) \right] . \end{aligned} \tag{7}$$

G tries to minimize the objective and D tries to maximize it. We add a ℓ_1 norm loss function to further constrain semantic information. The semantic loss is:

$$\mathcal{L}_{sem} = \sum\nolimits_{i=1}^{bs} \left\| \mathrm{G}_{X \to \hat{X}}(x_i; \theta_G) - \hat{x}_i \right\| . \tag{8}$$

To lessen the negative effect caused by parameter collapse, we follow CycleGAN to update D with a series of historical images rather than using new images from a separate epoch. In addition, we add a classification loss computed by

Policy	**Dataset**	**ResNet18**			**LeNet**			**ConvNet**		
		Sigmoid	ReLU	LReLU	Sigmoid	ReLU	LReLU	Sigmoid	ReLU	LReLU
ATS	MNIST	100	98.44	96.88	100	98.44	100	98.44	79.69	70.31
	F-MNIST	100	95.31	95.31	100	96.88	98.44	98.44	73.44	71.88
	CIFAR10	100	93.75	98.44	100	92.19	92.19	95.31	62.50	60.94
	SVHN	100	98.44	100	100	93.75	93.75	95.31	70.31	71.88
EtC	MNIST	100	100	100	100	95.31	93.75	93.75	98.44	96.88
	F-MNIST	100	96.88	96.88	100	96.88	96.88	100	95.31	92.19
	CIFAR10	100	96.88	96.88	100	96.88	96.88	100	92.19	85.94
	SVHN	100	93.75	95.31	100	96.88	96.88	100	85.94	87.50

Table 1: The LIA accuracy (%) of different model architectures and activation functions under the protection of ATS and EtC policy. We speculate that the reason for the low success rate in ConvNet is the MaxPool structure. LReLU: LeakyReLU.

C to G to better focus on categorical information in the distribution. Images generated by G are gradually aligned toward the target distribution during training. Then, we upload generator parameters to the central server to enable clients to complete FL. This also facilitates the training of global G because of more samples.

Local Classification Training

The feature extractor is introduced to achieve decoupling training in the non-overlapping and different images among clients. F is made up of an encoder Enc and a decoder Dec. We divide Enc into four blocks Enc_j, where $j \in \{1,2,3,4\}$. Each encoder block attempts to extract features, which are then restored to original dimensional pictures by the decoder. Let $\tilde{x}_i$ represents a generated image $\mathrm{G}_{X \to \hat{X}}(x_i; \theta_G)$. The objective of F is to extract efficient features from generated images by minimizing

$$\mathcal{L}_{fea} = \sum\nolimits_{i=1}^{bs} \|Dec(Enc(\tilde{x}_i)) - \tilde{x}_i\|^2. \tag{9}$$

Similarly, we use historical images to train F. Then, we initialize C and use the parameter as a key to guide the training of G with the $\mathcal{L}_{cls}$ computed by C. C is initialized with the key before each beginning of the global communication round. C receives final features from F and performs local update with n_c classes by minimizing

$$\mathcal{L}_{cls} = -\sum_{i=1}^{bs}\sum_{j=1}^{n_c} y_i\left(j\right) \log y_i'\left(j\right). \tag{10}$$

Because F converts encrypted images into efficient features, the classification network can be a fully connected neural network or a simple convolutional neural network.

Full Objective

Above all, our full objective of GAN is

$$\mathcal{L}_{GAN} = \mathcal{L}_{adv}^{(G,D)} + \lambda_{sem}\mathcal{L}_{sem} + \lambda_{cls}\mathcal{L}_{cls}, \tag{11}$$

where λ controls the importance of each loss term.

Experiments

Experiment Implementation

Datasets and Setup We carry out experiments on four datasets: MNIST (Deng 2012), FMNIST (Xiao, Rasul, and Vollgraf 2017), CIFAR10 (Krizhevsky and Hinton 2009), and SVHN (Netzer et al. 2011). Then we select one of the clients to act as a victim and carry out attacks. Our experiments are performed on the PyTorch platform using an NVIDIA GeForce 3090 Ti GPU.

Defense Baselines We compare our method with several defenses in federated learning: 1) ATS (Gao et al. 2021) suggests an automatic transformation search to find the ideal image transformation strategy; 2) EtC (Chuman, Sirichotedumrong, and Kiya 2019) utilizes a block-based image transformation method to encrypt images; 3) DP (Wei et al. 2020) clips gradients and adds Gaussian noise during training; 4) GC (Zhu, Liu, and Han 2019) performs privacy protection by pruning gradients; 5) FedCG (Wu et al. 2022) leverages conditional generative adversarial networks to guarantee privacy in federated learning. With the privacy budget ϵ/T in T global training epochs, we denote DP as DP<ϵ, C>, where C refers to the clipping bound. We also explore GC with different gradient compression degrees.

Attack Results

Results of Label Inference Attack We perform the attack (LIA) for 10 epochs under the protection of baselines (batch size is 64). In these baselines, the activation function is commonly ReLU or Sigmoid. We obtain the following observations from Table 1 and Table 2: 1) attack success rates are essentially 100% regardless of the complicated or simple models with Sigmoid; 2) rates are slightly lower, but most of them are more than 80% in other activation functions. One reason why Sigmoid performs better is that its results are all positive. When users upload classifier parameters, especially the last few fully connected layers, an attacker can effectively perform label inference, despite any protection measures they have taken. Compared with them, the attacker can't access penultimate gradients without sharing trained classifier parameters, making PPIDSG resist the attack.

Results of Membership Inference Attack We leverage test datasets as shadow datasets. A fully connected network with layer sizes of 10, 128, and 2 (the output layer) is applied as M_{attack}. We select the Adam optimizer and train the model for 100 epochs with a learning rate of 0.005. We start by comparing with ML-Leaks (Salem et al. 2019), which exploits a single model for an attack, and uses two different

	DP1	DP2	GC1	GC2	FedCG
Sigmoid	100	100	100	100	100
ReLU	90.63	92.19	89.06	90.63	82.81
LReLU	92.19	93.75	93.75	93.75	84.37

Table 2: The LIA accuracy (%) under other protection policies in LeNet with the CIFAR10 dataset. DP1: DP<5,10>, DP2: DP<20,5>, GC1: GC (10%), GC2: GC (40%). More results are presented in the Appendix.

Knowledge	MIA Accuracy (%)
Prediction vectors (ML-Leaks)	51.92
Prediction vectors (ours)	84.30
original images (MIA)	45.81

Table 3: Membership inference attack (MIA) accuracy using different data records and methods in the CIFAR10 dataset.

data types as inputs to M_{attack} in ATS. As illustrated in Table 3, it is difficult to attack with a single model. The observation demonstrates that our enhanced attack can improve the attack effect when overfitting is weakened in FL. Additionally, we discover that the uniform image distribution prevents the image knowledge from attacking effectively. Then we extend the attack to all defense strategies and assume that the attacker has a suspect dataset containing images from the victim and another user (part) or other users (all). Figure 3 presents that only EtC and our method obtain lower attack effects than other defenses. Uploading a trained classifier causes the exposure of overfitting, which further leads to privacy leakage. We speculate that EtC achieves MIA resistance by encrypting the original image distribution. Our proposal combines the above technique with the local classification model training to achieve privacy-preserving.

Results of Image Reconstruction Attack Due to the lack of full model parameters and loss terms, the adversary fails to undertake this attack in PPIDSG and can only recover images from the outputs of G. Figure 4 visually shows evaluations of image reconstruction attacks (RS). Compared to ATS and GC (10%), the attacker with other defense methods cannot recover original images visually. Moreover, we compare the privacy-preserving capability according to PSNR values between original and reconstructed images. The lower the PSNR score, the higher the privacy-preserving of this defense policy. We observe that PPIDSG achieves the lowest PSNR value in most datasets, implying the strongest privacy protection. Although the value is high in the SVHN dataset, the attacker cannot visually restore images.

By sharing classifier parameters to complete FL, not all baselines can protect against RS. Additionally, these methods also perform poorly in LIA and MIA. It indicates that sharing trained classifier parameters is a major privacy leakage in FL, and it may not be a suitable method for protection.

Defense Performance

Hyperparameter Configurations Our experiments are carried out on a FL system (McMahan et al. 2017) with ten clients, each of whom has the same amount of training data from the identical distribution. We set batch size as 64, image pool size as 10, and block sizes B_x and B_y in the encryption algorithm are 4. We apply an Adam optimizer and set the learning rate to 0.0002 in G and D. For F and C, we use a SGD optimizer and set the learning rate to 0.01 (weight decay is 0.001). Their initial learning rates are constant in the first 20 global iterations and then decrease linearly until they converge to 0. We set $\lambda_{sem} = 1$, $\lambda_{cls} = 2$ and run 50 global rounds. A random user is selected for accuracy testing since there is no global classification model in PPIDSG. More details can be found in the Appendix.

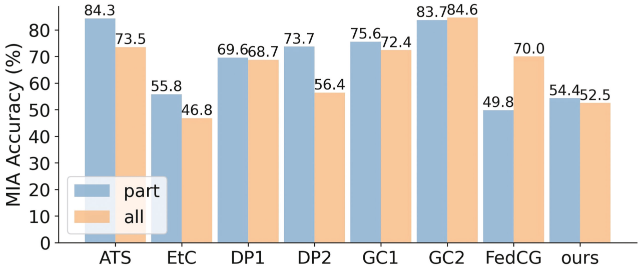

Figure 3: The MIA accuracy (%) of different victim data proportion with defense policies in the CIFAR10 dataset. Full dataset results are presented in the Appendix.

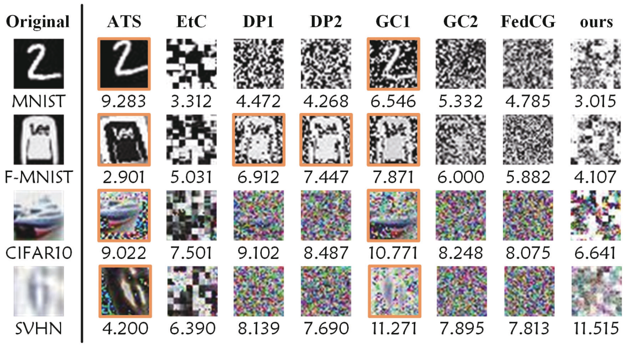

Figure 4: Qualitative and quantitative results (PSNR: dB) of RS. Full results are shown in the Appendix.

Results Table 4 verifies the highest model classification accuracy of our method compared with other techniques, achieving the best utility in all datasets within a tolerable time overhead (see Appendix), especially in color images with an accuracy exceeding 70%. In addition, the classifier architecture complexity in ATS and EtC, which we exploit in this paper as ResNet18, greatly influences the model utility, whereas our feature extractor improves model performance while reducing classifier complexity. In conclusion, our solution achieves competitive accuracy and outperforms others in terms of privacy-preserving.

To further investigate the model effectiveness of different modules, we conduct comparative studies as depicted in Figure 5. We have the following observations: 1) Compared with "original image" (benchmark) and "encrypted image" which take original images and encrypted images as inputs

Policy	Classification Accuracy (%)			
	MNIST	F-MNIST	CIFAR10	SVHN
ATS	98.96	89.23	59.67	85.22
EtC	98.06	89.41	53.34	78.70
DP1	97.42	85.58	49.29	82.70
DP2	97.54	85.01	44.43	80.28
GC1	97.61	85.81	54.07	84.36
GC2	97.22	85.09	50.91	79.96
FedCG	98.60	88.00	53.20	79.71
ours	**99.43**	**91.60**	**70.56**	**91.53**

Table 4: Classification accuracy results of test datasets. Full results are obtained in the Appendix.

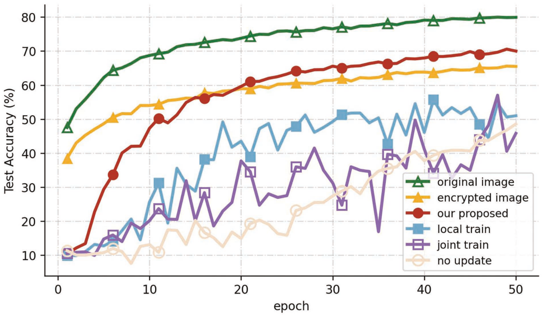

Figure 5: Comparison of defense accuracy under different modules in the CIFAR10 dataset.

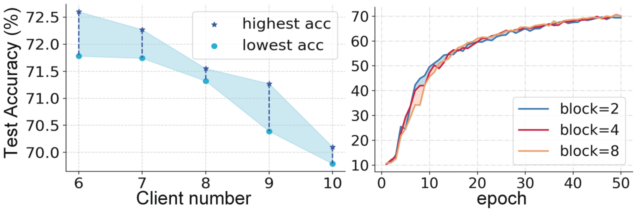

Figure 6: Test accuracy of different client numbers with all clients (left) and different block sizes (right). More results are presented in the Appendix.

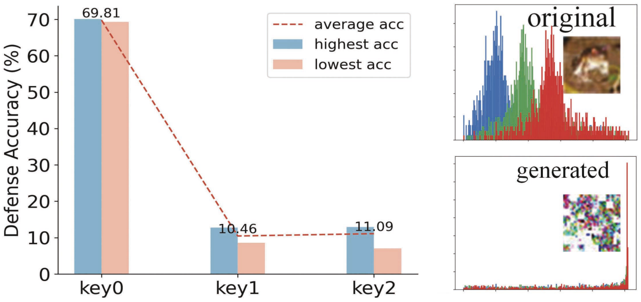

Figure 7: Left: Test accuracy (%) of classifier keys with all clients. Right: Pixel distribution between the original and generated images.

to F which jointly train with C respectively, our method has a slightly lower accuracy than the benchmark, exceeding the direct encryption policy. This indicates our PPIDSG can extract useful features to maintain model utility, demonstrating that merely sharing the image distribution can finish federated learning. 2) Our approach is more stable than “local train” and “joint train”, which train all models locally and jointly train F with C, respectively. It indicates our method can maintain a stable model while maintaining classification accuracy. Also, it explains why we train F and C independently and share the parameter of G. 3) We also notice that “no update”, which does not train the classifier and only uses the initial key, converges slowly. Therefore, we only employ the initial key at the beginning of each global round.

We also explore the following aspects: 1) the impact of different client numbers; 2) the effect of various image block sizes. As anticipated in Figure 6, the block size has almost no effect on model convergence, while the user number has a slight effect on classification accuracy.

Security Analyses

Classifier Key The parameter of C, which has a total of 37764106 parameters in CIFAR10 and SVHN datasets and 29899786 parameters in MNIST and F-MNIST datasets, is a crucial privacy factor in the testing process. We show how different parameters affect the test result by applying the corresponding $key0$ and two random keys. In the left of Figure 7, only $key0$ can perform valid testing while others cannot.

Target Image The image space in the target domain determines the difficulty of using an exhaustive attack. We formulate $N_{\mathrm{IR}}(n)$, $N_{\mathrm{IA}}(n)$, $N_{\mathrm{IF}}(n)$, and $N_{\mathrm{IC}}(n)$ as possible states of the encryption algorithm, and the target image space $N_{enc}(n)$ is represented by:

$$\begin{aligned} N_{enc}(n) &= N_{\mathrm{IR}}(n) \cdot N_{\mathrm{IA}}(n) \cdot N_{\mathrm{IF}}(n) \cdot N_{\mathrm{IC}}(n) \cdot n! \\ &= 4^n \cdot 2^n \cdot 3^n \cdot 6^n \cdot n!, \end{aligned} \tag{12}$$

where $n = (P_x \times P_y)/(B_x \times B_y)$, so that the target domain is a huge space that is difficult to decrypt.

Generated Image We analyze the histograms of the original and generated images on the right of Figure 7. Sensitive data is protected visually, and the pixel distribution is more uniform statistically. Particularly, blank areas of the generated image lead to abnormal distribution of final pixel areas, which we speculate can be compensated by improving the model aggregation and refining the image loss $\mathcal{L}_{sem}$.

Conclusion

We presented and supported the assumption that sharing trained classification model parameters is the main problem for privacy leakage in federated learning. To solve the problem, we subsequently designed a novel privacy-preserving method (PPIDSG) that combines a block scrambling-based encryption algorithm, an image distribution sharing method, and local classification training. Results showed that our scheme can successfully defend against attacks with high model utility. Future work will focus on: 1) improving the model aggregation and stable training; 2) enhancing users’ capacity to capture the image distribution.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China with Grant No. 62172383, No. 62231015, and No. 61802357, Anhui Provincial Key R&D Program with Grant No. S202103a05020098, Research Launch Project of University of Science and Technology of China with Grant No. KY0110000049.

References

Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H. B.; Mironov, I.; Talwar, K.; and Zhang, L. 2016. Deep learning with differential privacy. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 308–318.

Chuman, T.; Sirichotedumrong, W.; and Kiya, H. 2019. Encryption-then-compression systems using grayscale-based image encryption for JPEG images. *IEEE Transactions on Information Forensics and Security*, 14(6): 1515–1525.

Cubuk, E. D.; Zoph, B.; Mane, D.; Vasudevan, V.; and Le, Q. V. 2019. AutoAugment: Learning augmentation strategies from data. In *IEEE/CVF Conference on ComputerVision and Pattern Recognition (CVPR)*, 113–123.

Deng, L. 2012. The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6): 141–142.

Ding, Y.; Wu, G.; Chen, D.; Zhang, N.; Gong, L.; Cao, M.; and Qin, Z. 2021. DeepEDN: A deep-learning-based image encryption and decryption network for internet of medical things. *IEEE Internet of Things Journal*, 8(3): 1504–1518.

Duan, J.; Kong, F.; Wang, S.; Shi, X.; and Xu, K. 2023. Are diffusion models vulnerable to membership inference attacks? arXiv:2302.01316.

Fu, C.; Zhang, X.; Ji, S.; Chen, J.; Wu, J.; Guo, S.; Zhou, J.; Liu, A. X.; and Wang, T. 2022. Label Inference Attacks Against Vertical Federated Learning. In *Proceedings of the 31st USENIX Security Symposium*, 1397–1414.

Gao, W.; Guo, S.; Zhang, T.; Qiu, H.; Wen, Y.; and Liu, Y. 2021. Privacy-preserving collaborative learning with automatic transformation search. In *IEEE/CVF Conference on ComputerVision and Pattern Recognition (CVPR)*, 114–123.

Geng, J.; Mou, Y.; Li, F.; Li, Q.; Beyan, O.; Decker, S.; and Rong., C. 2022. Towards general deep leakage in federated learning. In *International Workshop on Trustable, Verifiable and Auditable Federated Learning in Conjunction with AAAI*.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio., Y. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2672–2680.

Guo, P.; Wang, P.; Zhou, J.; Jiang, S.; and Patel, V. M. 2021. Multi-institutional collaborations for improving deep learning-based magnetic resonance image reconstruction using federated learning. In *IEEE/CVF Conference on ComputerVision and Pattern Recognition (CVPR)*, 2423–2432.

Hayes, J.; Melis, L.; Danezis, G.; and Cristofaro, E. D. 2019. LOGAN: Membership Inference Attacks Against Generative Models. *Proceedings on Privacy Enhancing Technologies.*

Hitaj, B.; Ateniese, G.; and Perez-Cruz, F. 2017. Deep models under the GAN: Information leakage from collaborative deep learning. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 603–618.

Huang, Y.; Song, Z.; Li, K.; and Arora, S. 2020. InstaHide: Instance-hiding schemes for private distributed learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 4457–4468.

Jiang, M.; Wang, Z.; and Dou, Q. 2022. HarmoFL: Harmonizing Local and Global Drifts in Federated Learning on Heterogeneous Medical Images. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, 914–922.

Jin, W.; Yao, Y.; Han, S.; Joe-Wong, C.; Ravi, S.; Avestimehr, S.; and He, C. 2023. FedML-HE: An Efficient Homomorphic-Encryption-Based Privacy-Preserving Federated Learning System. arXiv:2303.10837.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images.

Liao, X.; Liu, W.; Zheng, X.; Yao, B.; and Chen, C. 2023. PPGenCDR: A Stable and Robust Framework for Privacy-Preserving Cross-Domain Recommendation. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI)*, 4453–4461.

Liu, Q.; Chen, C.; Qin, J.; Dou, Q.; and Heng, P.-A. 2021. FedDG: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In *IEEE/CVF Conference on ComputerVision and Pattern Recognition (CVPR)*, 1013–1023.

McMahan, H. B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 1273–1282.

Melis, L.; Song, C.; Cristofaro, E. D.; and Shmatikov, V. 2019. Exploiting Unintended Feature Leakage in Collaborative Learning. In *IEEE Symposium on Security and Privacy*, 691–706.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*.

Phong, L. T.; Aono, Y.; Hayashi, T.; Wang, L.; and Moriai, S. 2018. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 13(5): 1333–1345.

Salem, A.; Zhang, Y.; Humbert, M.; Berrang, P.; Fritz, M.; and Backes, M. 2019. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *26th Annual Network and Distributed System Security Symposium (NDSS)*.

Sellami, A.; and Tabbone, S. 2022. Deep neural networks-based relevant latent representation learning for hyperspectral image classification. *Pattern Recognition*, 121: 108224.

Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy (SP)*, 3–18.

Sun, J.; Li, A.; Wang, B.; Yang, H.; Li, H.; and Chen, Y. 2021. Soteria: Provable defense against privacy leakage in federated learning from representation perspective. In *IEEE/CVF Conference on ComputerVision and Pattern Recognition (CVPR)*, 9307–9315.

Wei, K.; Li, J.; Ding, M.; Ma, C.; Yang, H. H.; Farokhi, F.; Jin, S.; Quek, T. Q. S.; and Poor, H. V. 2020. Federated Learning With Differential Privacy: Algorithms and Performance Analysis. *IEEE Transactions on Information Forensics and Security*, 15: 3454–3469.

Wu, Y.; Kang, Y.; Luo, J.; He, Y.; Fan, L.; Pan, R.; and Yang, Q. 2022. FedCG: Leverage Conditional GAN for Protecting Privacy and Maintaining Competitive Performance in Federated Learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2334–2340.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. arXiv:1708.07747.

Xie, K.; Zhang, Z.; Li, B.; Kang, J.; Niyato, D.; Xie, S.; and Wu, Y. 2022. Efficient Federated Learning With Spike Neural Networks for Traffic Sign Recognition. *IEEE Transactions on Vehicular Technology*, 71(9): 9980–9992.

Yin, H.; Mallya, A.; Vahdat, A.; Alvarez, J. M.; Kautz, J.; and Molchanov, P. 2021. See through gradients: Image batch recovery via gradinversion. In *IEEE/CVF Conference on ComputerVision and Pattern Recognition (CVPR)*, 16332–16341.

Yu, Y.; Liu, Q.; Wu, L.; Yu, R.; Yu, S. L.; and Zhang, Z. 2023. Untargeted Attack against Federated Recommendation Systems via Poisonous Item Embeddings and the Defense. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI)*, 4854–4863.

Zhao, B.; Mopuri, K. R.; and Bilen, H. 2020. idlg: Improved deep leakage from gradients. arXiv:2001.02610.

Zhou, M.; Wu, J.; Liu, Y.; Liu, S.; and Zhu, C. 2020. DaST: Data-free substitute training for adversarial attacks. In *IEEE/CVF Conference on ComputerVision and Pattern Recognition (CVPR)*, 231–240.

Zhu, J.; Yao, R.; and Blaschko, M. B. 2023. Surrogate Model Extension (SME): A Fast and Accurate Weight Update Attack on Federated Learning. In *International Conference on Machine Learning (ICML)*.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2242–2251.

Zhu, L.; Liu, Z.; and Han, S. 2019. Deep leakage from gradients. In *Advances in Neural Information Processing Systems (NeurIPS)*.