Privacy-Preserving Federated Compressed Learning Against Data Reconstruction Attacks Based on Secure Data

Di Xiao(✉), Jinkun Li, and Min Li

College of Computer Science, Chongqing University, Chongqing 400044, China
{dixiao,jkli,minli}@cqu.edu.cn

Abstract. Federated learning is a new distributed learning framework with data privacy preserving in which multiple users collaboratively train models without sharing data. However, recent studies highlight potential privacy leakage through shared gradient information. Several defense strategies, including gradient information encryption and perturbation, have been suggested. But these strategies either involve high complexity or are susceptible to attacks. To counter these challenges, we propose to train on secure compressive measurements by compressed learning, thereby achieving local data privacy protection with slight performance degradation. A feasible method to boost performance in compressed learning is the joint optimization of the sampling matrix and the inference network during the training phase, but this may suffer from data reconstruction attacks again. Thus, we further incorporate a traditional lightweight encryption scheme to protect data privacy. Experiments conducted on MNIST and FMNIST datasets substantiate that our schemes achieve a satisfactory balance between privacy protection and model performance.

Keywords: Federated learning · Data reconstruction attack · Privacy-preserving · Compressed learning · Lightweight encryption

1 Introduction

Artificial Intelligence (AI) is a transformative technology that enables machines to mimic human-like thinking, learning, and reasoning capabilities. The performance of AI systems heavily relies on the availability of a substantial amount of high-quality data. However, due to growing public concern over privacy issues and increasingly stringent data protection regulations, collecting training data that may contain private information has become a significant challenge. To tackle these challenges, federated learning (FL) [9] has emerged as a promising solution. FL offers a powerful approach to addressing the issues of "data islands" and privacy, making it a prominent area of research in the field of AI technology. Unlike traditional centralized machine learning training, FL aggregates model parameter updates from local devices to a central server, so as to realize model training and updating without sharing raw data.

B. Luo et al. (Eds.): ICONIP 2023, CCIS 1969, pp. 325–339, 2024.
https://doi.org/10.1007/978-981-99-8184-7_25

Unfortunately, this approach falls short in guaranteeing data privacy sufficiently. Numerous studies have demonstrated the ability to infer sensitive information from the shared gradient information. Membership inference attack [12] is the pioneering research that reveals privacy leakage in FL, by analyzing whether a specific sample was present in the training data. Building upon this, subsequent research has shown that user-level information can be reconstructed from the gradient [16]. Later, the data reconstruction attack that recovers the original data through the gradient is produced, fundamentally challenging the notion that FL adequately protects local data privacy. In this paper, we focus on data reconstruction attacks, which represent a type of inference attack.

Recently, there have been several data reconstruction attack methods [4,6, 18–20] proposed, which have demonstrated the ability of an attacker to reconstruct the local training data by exploiting the shared gradient. These methods primarily achieve data reconstruction by minimizing the distance between the gradient generated by the virtual data and label, and the gradient generated by the real data and label. To tackle the problem of data reconstruction attacks, various defense strategies have been proposed. These strategies can be broadly classified into two categories. The first category includes cryptographic-based methods, such as multiparty secure computation. These methods employ secure aggregation protocols, such as homomorphic encryption [1,8,13], to safeguard the original gradient information from exposure. The second category involves gradient perturbation methods, with differential privacy [15,17] being a prominent example. Furthermore, [20] proposes that gradient compression can effectively mitigate gradient privacy concerns. Another defensive approach suggested by Sun et al. [14], called Sotera, involves perturbing the data before gradient calculation. Unfortunately, the method proposed in [6] can reconstruct similar data even when gradient perturbation strategies such as gradient clipping, additive noise to gradient, gradient sparsification, and Sotera are employed.

Li et al. [6] propose to utilize the latent space of a generative adversarial network (GAN) trained on a public image dataset to compensate for information loss caused by gradient degradation. However, training on a dataset that is not relevant to the public image dataset can prevent GAN from learning relevant prior information. Hence, we are inspired to address the data reconstruction attack problem in FL by using secure data. Compressed sensing (CS) [3] is a mathematical framework for efficient signal acquisition and robust recovery. In addition to compression, CS measurements are encrypted. Calderbank et al. [2] introduce the concept of compressed learning (CL), where the inference system is directly built on CS measurements. Subsequent researches [7,10,11,21] have further improved CL, with [10] demonstrating that CL can achieve performance almost comparable to the original image domain. To the best of our knowledge, CL has not yet been explored in the context of FL to address data reconstruction attacks, making it a promising area for investigation.

In this paper, we design a privacy-preserving framework for FL based on CL, which exploits the secrecy property of CS measurements to achieve the defensive effect against data reconstruction attacks. Previous studies [10,21] have shown that jointly optimizing the sampling matrix and the inference network can improve model performance. However, as the sampling matrix is part of the model during the training process, it remains vulnerable to data reconstruction attacks. To address this issue, we introduce traditional cryptography to encrypt the training data, which protects the data from the threat of data reconstruction attacks. Different from the first two categories of defense methods based on gradient encryption and gradient perturbation, our approach focuses on training with secure data and is orthogonal to the previous methods. Experimental results further validate the feasibility of our proposed method.

Our main contributions can be summarized as follows:

- We propose to apply CL to address the privacy leakage problem caused by data reconstruction attacks in FL.
- To further enhance performance, we jointly optimize the sampling matrix and the inference network. Additionally, we introduce lightweight encryption novel methods in both the spatial and frequency domains to protect the training data.
- Experimental results demonstrate the effectiveness of our proposed scheme in achieving satisfactory privacy and utility.

2 Related Work

2.1 Data Reconstruction Attacks

Recently, Zhu et al. [20] propose to reconstruct the original data and label by making the ℓ_2 norm of the gradient of a pair of virtual data and label close to that of the gradient generated by the real data and label. A follow-up work [19] proposes to use the gradient of the last fully connected network layer to recover the true label. However, these approaches encounter difficulties when applied to large-scale and discontinuous network models. Then Geiping et al. [4] introduce a recovery method that is independent of the gradient magnitude. They utilize cosine similarity to measure the similarity between gradients, demonstrating that more complex data can be recovered even in deeper models. Later, Yin et al. [18] introduce a group consistency regularization term, enabling the reconstruction of a large batch of images after gradient averaging.

2.2 Defenses Against Data Reconstruction Attacks

The existing defense schemes for FL can be classified into two categories, one is based on gradient encryption, and the other is based on gradient perturbation.

Homomorphic encryption serves as an effective secure aggregation scheme for gradient protection. In [1], an asynchronous stochastic gradient descent scheme leveraging additive homomorphic encryption is proposed. To address the high computational complexity of homomorphic encryption aggregation on the server side, [13] introduces a calculation provider third party. Additionally, [8] proposes homomorphic encryption with multiple keys to resist collusion attacks. However, these methods often impose considerable computational complexity, rendering them unsuitable for resource-constrained users.

Another line of research explores perturbing the gradient to degrade it, thereby preventing privacy leaks. Differential privacy, which introduces random noise, can quantify and limit the disclosure of user privacy. The convergence of FL with the introduction of differential privacy is proven in [17]. Moreover, [15] incorporates local differential privacy into FL, and customizes improvements suitable for FL. While these gradient perturbation methods offer lower computational complexity, they still result in a decrease in model accuracy.

2.3 Compressed Sensing

Compressed sensing (CS) [3] is a mathematical paradigm that exploits signal redundancy to accurately reconstruct signals from a significantly reduced number of measurements compared to the Shannon sampling rate.

Specifically, given a signal $x \in \mathbb{R}^N$, the CS measurement $y \in \mathbb{R}^N$ is obtained by the sampling process $y = \mathit{\Phi} x$, where $\mathit{\Phi} \in \mathbb{R}^{M \times N}$ is the sampling matrix, $M \ll N$, and the sampling rate is defined as $\gamma = M/N$. Since the number of unknowns is much larger than the number of equations, it is often impossible to reconstruct x from the observations y. But if x is sparse in some basis $\mathit{\Psi}$, then x can be reconstructed from y, which is CS theory.

There are many studies devoted to solving the above reconstruction problem, and traditional optimization algorithms usually reconstruct the original signal x by solving the following optimization problem:

$$\min_{x} ||\mathit{\Psi} x||_1 \quad s.t. \;\; y = \mathit{\Phi} x, \tag{1}$$

where $\mathit{\Psi}$ denotes some sparse basis on which x is sparse.

2.4 Compressed Learning

Compressed learning (CL) aims to perform inference tasks directly in the measurement domain without signal recovery. The concept is initially proposed by Calderbank et al. [2], who provide the first theoretical foundation for CL and demonstrate the feasibility of performing inference tasks in the compressed domain. Subsequently, Lohit et al. [7] utilize Convolutional Neural Network (CNN) for classification in the compressed domain. Building upon this, Adler et al. [21] devise an end-to-end deep learning solution for classification in the compressed domain, jointly optimizing the sampling matrix and inference operator. In [11], the authors demonstrate the robustness of the CL scheme by performing inference using partially observed image data. Recently, [10] employs an elaborate transformer network to conduct inference tasks in the measurement domain.

3 Methodology

In our approach, we utilize a CL scheme to project the measurement back into the image space, generating a noise-like proxy image in Sect. 3.1. This proxy image is directly used as the training data for the FL client to resist data reconstruction attacks. To further enhance model performance, we jointly optimize the sampling matrix and the subsequent inference network. However, during training, the original training data is re-input into the model, which will suffer from data reconstruction attacks. To address this, we propose encrypting the training data before training. In Sect. 3.2, we introduce a spatial domain encryption scheme. Additionally, to enhance the security of the encryption scheme, we propose a frequency domain encryption scheme in Sect. 3.3.

3.1 Training with Proxy Image

Motivation. In the scenario where we possess CS measurements, obtained through an imaging device like a single-pixel camera, these measurements often manifest as noise distributions. We can adopt the framework of CL and use the secure measurements as the training data of the client in FL, so that an honest but curious server is unable to extract any information about the original image, even if it obtains the measurements through data reconstruction attack methods. Furthermore, let us assume that the server has acquired our sampling matrix through some means, enabling them to potentially reconstruct the original image using CS reconstruction algorithms. However, it is essential to note that the distribution of CS measurements is closely linked to the image and is not identical, even when the same sampling matrix is applied. Therefore, before training the network model, we normalize all measurements, as this step is a crucial part of the neural network training process. Consequently, even if the normalized measurements are inferred through a data reconstruction attack, the inability to restore them to their original distribution prevents the recovery of the corresponding original images using CS reconstruction algorithms.

We propose a CL scheme to resist the data reconstruction attack and apply Fig. 1 to illustrate the whole process of the client during one communication round. For an image, we flatten it into a one-dimensional vector and obtain the measurement by CS sampling. The measurement obtained is a one-dimensional vector, which is usually projected back to the original image dimension and then used by CNN to extract features for image classification tasks. To restore the measurement back to the original image dimension, we can employ a matrix related to the sampling matrix (such as its transpose) or utilize a fully connected network layer. Consider that the data reconstruction attack method we used in the subsequent experiments is more effective for recovering image data, because of the addition of the total variation regularization term in its loss function. Therefore, we project the measurement back into the image space using the transpose of the sampling matrix and reshape it to obtain the proxy image. Subsequently, the proxy image undergoes normalization to ensure consistency in the gray value interval [0, 255]. This step is necessary due to significant changes in the pixel distribution of the proxy image. The normalized proxy image is then employed as training data for the network model. The server can only perform data reconstruction attacks using shared gradients to recover the proxy image, but can not obtain information about the original image.

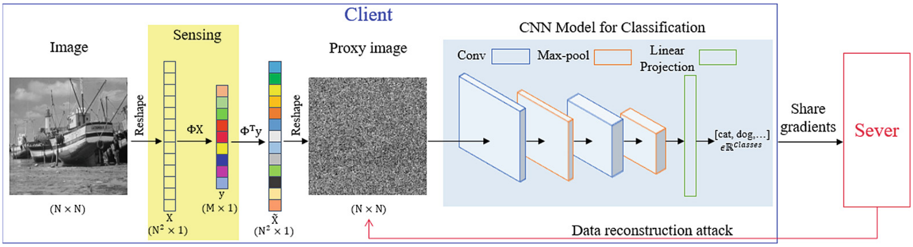

Fig. 1. The whole process of client local training when taking one picture as an example. Red arrows represents the data reconstruction attack. (Color figure online)

- Step 1: The client performs CS on the local dataset D. For one $N \times N$ image from D, it is first flattened into a one-dimensional vector. Then the flattened image is sampled by $y = \Phi X$ to obtain the measurement y, where $\Phi \in \mathbb{R}^{M \times N}$ and the sampling rate is M/N.
- Step 2: Next, the one-dimensional vector projected back into the image space is obtained by $\widetilde{X} = \Phi^T y$, and the proxy image is obtained by reshaping it to $N \times N$.
- Step 3: The proxy image is normalized before being fed into the CNN model for image classification task training.
- Step 4: After the local training, the selected clients of this communication round upload the model updates to the server.
- Step 5: The server aggregates all the received model updates and transmits them back to all clients.

3.2 Training with Encrypted Data in the Spatial Domain

The scheme in Sect. 3.1 uses a fixed sampling matrix for all the training data. However, optimizing the sampling matrix with the subsequent inference network concurrently will further improve the network's inference performance. To achieve this, the original training data needs to be fed into the network during the training process. However, an honest but curious server could potentially utilize a data reconstruction attack method to access the original training data.

Therefore, in this section, we propose a novel approach to counter the data reconstruction attack by encrypting the image data prior to training in the neural network. However, it is crucial to strike a balance between usability and privacy, as excessive encryption of the images can render them untrainable. To address this concern, we explore a two-dimensional random permutation scheme for encrypting the image data.

For a two-dimensional image $X \in \mathbb{R}^{N \times N}$, multiplying the left and right sides of it by a row random permutation matrix P and a column random permutation matrix Q, respectively, and the encryption process can be expressed as

$$E(X) = PXQ. \tag{2}$$

The matrices P and Q are $N \times N$ square matrices with only one '1' in each row and column and zeros in the rest. We can use a chaotic system such as Logistic map to generate these two square matrices. By controlling the keys k1 and k2 as the initial values of the chaotic system, Algorithm 1 describes the process of generating P and Q using Logistic chaotic map. We apply Fig. 2 to illustrate the whole process of the client during one communication round.

Algorithm 1: Logistic chaotic system generates two random permutation matrices.

```
Input: Initial value $x_0$,$y_0$;Number of iterations $N$;Control parameter
       $\mu = 3.56995$.
Initial chaotic sequence $S1 = \{\}; S2 = \{\}$;
for $i = 1$ to $i = N$ do
    $x_i = \mu \times x_{i-1} \times (1 - x_{i-1})$;
    $y_i = \mu \times y_{i-1} \times (1 - y_{i-1})$;
    $S1 = S1 \bigcup x_i$;
    $S2 = S1 \bigcup y_i$;
Sort $S1$ and $S2$ separately;
Denote their corresponding position indexes in the new ordered sequences as
$indexS1$ and $indexS2$, which are in the range of $[1, N]$;
$P_{i,j} = \begin{cases} 1, & i = indexS1(j), \\ 0, & others \end{cases}, Q_{i,j} = \begin{cases} 1, & j = indexS2(i), \\ 0, & others \end{cases}$;
Output: Row random permutation matrix $P$; Column random permutation
        matrix $Q$.
```

- Step 1: The client encrypts each $N \times N$ image from the local dataset D, using the two-dimensional random permutation.
- Step 2: The encrypted images are normalized and then input into the sensing module.
- Step 3: The sensing module samples the flattened one-dimensional vector of each image with a learnable sampling matrix to obtain the measurements.
- Step 4: The measurements are first activated by ReLU and then projected back to the one-dimensional vector of the original image dimension using a fully connected network layer. The vector is reshaped to be input into the subsequent CNN module.
- Step 5: After the local training, the selected clients of this communication round upload their model updates to the server.
- Step 6: The server aggregates all the received model updates and transmits them back to all clients.

3.3 Training with Encrypted Data in the Frequency Domain

To enhance the security and resistance against statistical analysis and other attacks, it is beneficial to convert the image into a transform domain, such

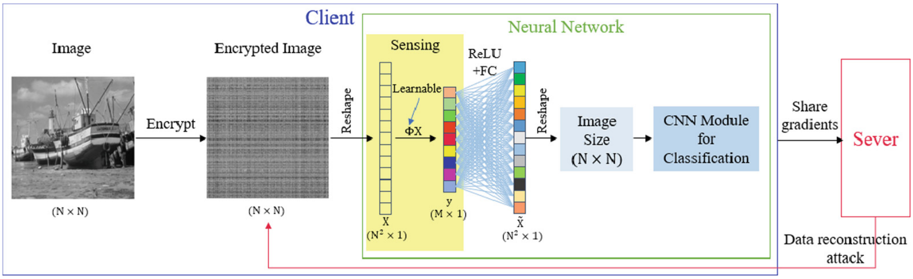

Fig. 2. The whole process of client local training when taking one picture as an example. Red arrows represents the data reconstruction attack. (Color figure online)

as the frequency domain, before encrypting it. Therefore, in this section, we propose a two-dimensional random permutation encryption scheme for the two-dimensional discrete cosine transform coefficients of the image.

For a two-dimensional image $X \in \mathbb{R}^{N \times N}$, we first apply the two-dimensional discrete cosine transform (2D-DCT) on it and obtain a transformed coefficient matrix $D \in R^{N \times N}$. This process can be expressed as

$$D = \varPsi X \varPsi^T, \tag{3}$$

where $\varPsi$ represents the sparse basis matrix of the discrete cosine transform. Next, the coefficient matrix D is encrypted using the two-dimensional random permutation scheme described in Sect. 3.2. This encryption process results in the encrypted coefficient matrix E. Finally, the encrypted coefficient matrix E is transformed back to the image domain using the two-dimensional inverse discrete cosine transform. The process can be expressed as

$$E = PDQ, \tag{4}$$

$$X^{'} = \varPsi^T E \varPsi, \tag{5}$$

where P and Q represent the row random permutation matrix and column random permutation matrix, respectively, $X^{'}$ denotes the encrypted image after transforming back to the image domain. Due to the scrambling of the 2D-DCT coefficient matrix, the distribution range of pixel values undergoes significant changes after the shuffled coefficient matrix is transformed back to the image domain. Therefore, after encrypting, all encrypted images are normalized to the same gray value interval [0, 255], which further improves the security of the frequency domain encryption scheme. The overall client process represented by this scheme can also refer to Fig. 2.

4 Experiment

In this section, we begin by introducing our experimental settings in Sect. 4.1. The data reconstruction attack results on different datasets are presented in

Sect. 4.2. We conduct a security performance evaluation in Sect. 4.3, followed by the presentation of model performance results in Sect. 4.4. For ease of reference, we assign the names Scheme I, Scheme II, and Scheme III to the schemes discussed in Sects. 3.1, 3.2, and 3.3, respectively.

4.1 Experimental Settings

Datasets and Evaluation Metrics. We use the MNIST and FMNIST datasets for our experiments. These two datasets are commonly used image classification datasets and are widely used for deep learning algorithms. Our evaluation focuses on defense and performance. To assess the effectiveness of defense techniques, we employ the peak signal-to-noise ratio (PSNR) as a metric. A lower PSNR value indicates a greater visual difference. In terms of performance, we evaluate the test accuracy of model across different datasets.

Training Details. In our simulation experiments, the FL system consists of a central server and one hundred clients. The total number of training rounds is 100, and the local epoch in each round is 5. In each communication round, 50 clients are randomly selected to participate in FL training. For the MNIST dataset, we use the LeNet5 model [5], and the local batch size is set to 32. For the FMnist dataset, we use a small ConvNet model which contains three convolutional layers and a fully connected layer. The number of channels in the convolutional layer is 16,32,64, respectively, and the size of the convolution kernel is 5. The ReLU activation function and Max pooling are used after each convolution, and the local batch size is set to 64. We train our models on PyTorch using a 1080Ti card, and all models are optimized using SGD optimizer with momentum set to 0.9. The learning rate is initially set to 0.01, and the learning rate is reduced to 0.001 after 30 rounds of communication.

4.2 Defense Effect Against Data Reconstruction Attacks

We apply the Inverting Gradients (IG) method proposed in [4] to recover a single input image. The recovery process consists of 24,000 iterations, with a total variation term weight of 0.0001. To ensure the independence of recovery results from the initial seed, we conduct 10 repetitions of the experiment for each attack and report the result with the lowest loss.

Figure 3 displays the results of IG attack against the MNIST and FMNIST datasets. In the original FL scenario without privacy protection, the original data is fully exposed. However, in Scheme I, the recovered image is only a proxy image, which resembles the noise distribution, thereby making its recovery more challenging. In Scheme II, it can be observed that the recovered image is relatively clear. However, since the image itself is encrypted, the obtained image remains an unrecognizable encrypted form. Scheme III demonstrates that the recovered image has no correlation with the original image, indicating that the encryption effect is superior to that of Scheme II.

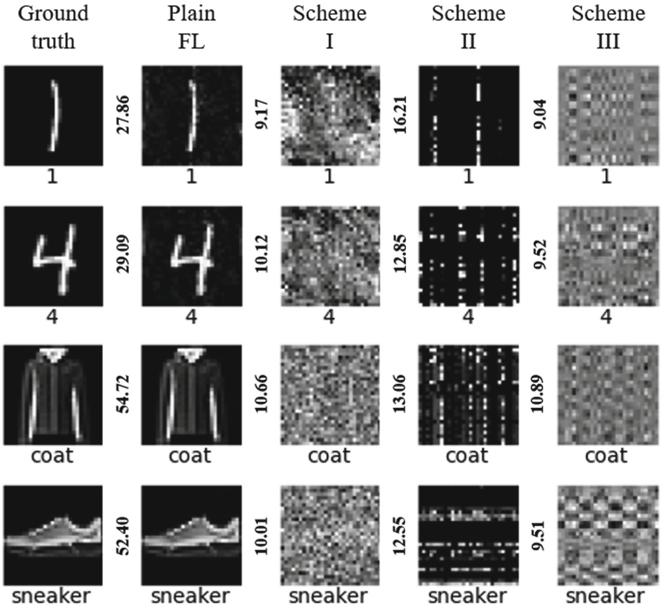

Fig. 3. Recovery results of IG attack on the MNIST and FMNIST datasets for different methods. For our schemes, the sampling rate is set to 0.1. The PSNR between the image and its ground truth is displayed on the left side of the image.

We also conduct IG attack experiments on the gradient compression method proposed in [20]. The defense effect under different sparsity ratios P is illustrated in Fig. 4. It is observed that the image remains recognizable until P = 0.01. It is worth noting that [20] claims resistance against data reconstruction attacks when the sparsity ratio approaches 0.8. However, while [20] utilizes a reconstruction attack based on minimizing the ℓ_2 norm between gradients, our paper employs IG attack based on minimizing the cosine similarity between gradients.

4.3 Secure Performance Evaluation

In Scheme I, we employ a linear transformation operation on the image. It is important to consider that if the CS sampling matrix is stolen, an attacker may utilize the least square method to reconstruct the original image based on the proxy image recovered through the data reconstruction attack, this process can be described as

$$\hat{X} = (\Phi^T \Phi)^{-1} X, \tag{6}$$

where $X \in \mathbb{R}^{N \times N}$ represents the proxy image recovered through the data reconstruction attack, and $\hat{X}$ denotes the final original image reconstructed using the least square method. After projecting the CS measurement back into the original image space, the proxy image is then normalized to the gray value interval of [0, 255]. So, if the pixel distribution of the proxy image is not restored, the attacker

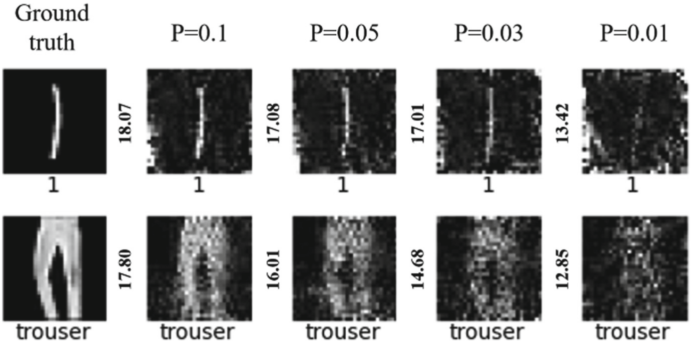

Fig. 4. Recovery results of IG attack at different sparsity ratios. The PSNR between the image and its ground truth is displayed on the left side of the image.

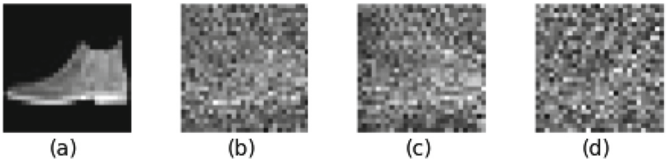

Fig. 5. (a) original image, (b) proxy image with a sampling rate of 0.25, (c) image inferred from the IG attack, (d) image reconstruct from the inferred image.

cannot obtain a relatively clear image using the least square method. This claim is further validated by the example depicted in Fig. 5.

In Scheme II and Scheme III, we employ the two-dimensional random permutation encryption method. In Scheme II, the encryption is applied to the original image, while in Scheme III, it is applied to the corresponding 2D-DCT coefficients of the original image. The key space for an $N \times N$ dimensional image is $(N!)^2$, which provides sufficient resistance against brute-force search attacks. It is worth noting that even if the encryption key is compromised, Scheme III remains secure due to the normalization of all encrypted images to the same gray value interval [0, 255].

We employ information entropy to evaluate the level of chaos in the datasets associated with the three schemes. Given the strong correlation between adjacent pixels, image data tends to exhibit low entropy, indicating higher predictability. As depicted in Table 1, the corresponding datasets of Scheme I and Scheme III demonstrate relatively higher entropy, rendering the data less predictable. In Scheme II, only the pixel positions in the image are shuffled, while the distribution of pixel values remains unchanged, resulting in the same entropy as the original dataset. Consequently, Scheme III significantly enhances the security performance compared to Scheme II.

Table 1. The entropy of dataset corresponding to different schemes.

Dataset	Original	Scheme I	Scheme II	Scheme III
MNIST	1.60	7.13	1.60	6.89
FMNIST	4.12	7.19	4.12	7.06

4.4 Model Performance

The accuracy of the model on different test datasets for various compression sampling rates and schemes is presented in Table 2 and Table 3. For Scheme I, the model demonstrates improved performance with higher compression sampling rates. At a sampling rate of 0.25, the model performs reasonably well, slightly below the federated average (FedAvg) algorithm without protection. Turning to Scheme II, employing only encryption without a sensing module at a sampling rate of 1, an interesting observation emerges. When a sensing module is added, the final model achieves higher accuracy compared to the scheme with encryption alone, particularly at sampling rates of 0.25, 0.1, and 0.05. One potential explanation is that the addition of CS reduces the dimensionality of the data, rendering the data distribution more traceable and improving the model's performance. Scheme II outperforms Scheme I across all sampling rates. Introducing Scheme III to further protect the training data, we observe that, similar to Scheme II, the model's performance with encryption followed by CS surpasses that of the model with encryption alone (sampling rate of 1) at sampling rates of 0.25 and 0.1. Both Scheme II and Scheme III exhibit relatively minor performance degradation compared to the unprotected FedAvg algorithm.

In other privacy-preserving approaches in FL, the adoption of gradient encryption methods, such as homomorphic encryption, significantly increases computational complexity and communication overhead. On the other hand, gradient perturbation methods such as differential privacy approaches, often lack specific demonstrations of their protective effects and primarily focus on different privacy budget scenarios. For instance, in the study conducted by [15], the optimal test accuracy on the FMNIST dataset is reported as 86.93%, whereas the FedAvg algorithm achieves approximately 90% accuracy. Notably, some of our proposed schemes achieve comparable results at an appropriate sampling rate.

We evaluate the performance of the gradient compression (GC) method proposed in [20] with a prune ratio of 0.99. The accuracy achieved by GC on the MNIST and FMNIST test datasets is 96.78% and 85.74%, respectively. These results indicate that our schemes all outperform GC at some sampling rates.

Table 2. Model accuracy of different schemes on the MNIST test dataset.

Sampling Rate	Scheme I	Scheme II	Scheme III	FedAvg
1	–	96.30	96.23	98.98
0.25	97.53	97.57	96.84	
0.1	95.32	97.34	96.53	
0.05	91.55	96.94	95.96	
0.01	56.68	93.45	93.05	

Table 3. Model accuracy of different schemes on the FMNIST test dataset.

Sampling Rate	Scheme I	Scheme II	Scheme III	FedAvg
1	–	87.26	87.54	89.89
0.25	86.69	87.98	87.98	
0.1	85.32	87.88	87.72	
0.05	83.68	87.55	87.43	
0.01	58.69	85.27	84.57	

5 Conclusion

In this paper, we propose a novel privacy-preserving FL framework based on CL. Our approach introduces CL as a mechanism to address gradient leakage privacy concerns in FL, and we demonstrate its feasibility. Additionally, we propose the utilization of lightweight encrypted data as a protective scheme against data reconstruction attacks. Through simulation results, we validate the effectiveness of our schemes in resisting attacks, with slight impact on accuracy under suitable compression rates. In future research, we plan to explore additional protection schemes derived from CL for integration into FL. Furthermore, we aim to develop protection solutions tailored for resource-constrained device scenarios, ensuring their suitability and practicality.

Acknowledgements. The work was supported by the National Key R&D Program of China under Grant 2020YFB1805400, the National Natural Science Foundation of China under Grant 62072063 and the Project Supported by Graduate Student Research and Innovation Foundation of Chongqing, China under Grant CYB22063.

References

1. Aono, Y., Hayashi, T., Wang, L., Moriai, S., et al.: Privacy-preserving deep learning via additively homomorphic encryption. IEEE Trans. Inf. Forensics Secur. **13**(5), 1333–1345 (2017)
2. Calderbank, R., Jafarpour, S., Schapire, R.: Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain. preprint (2009)

3. Donoho, D.L.: Compressed sensing. IEEE Trans. Inf. Theory **52**(4), 1289–1306 (2006)
4. Geiping, J., Bauermeister, H., Dröge, H., Moeller, M.: Inverting gradients-how easy is it to break privacy in federated learning? Adv. Neural. Inf. Process. Syst. **33**, 16937–16947 (2020)
5. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
6. Li, Z., Zhang, J., Liu, L., Liu, J.: Auditing privacy defenses in federated learning via generative gradient leakage. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10132–10142 (2022)
7. Lohit, S., Kulkarni, K., Turaga, P.: Direct inference on compressive measurements using convolutional neural networks. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 1913–1917. IEEE (2016)
8. Ma, J., Naas, S.A., Sigg, S., Lyu, X.: Privacy-preserving federated learning based on multi-key homomorphic encryption. Int. J. Intell. Syst. **37**(9), 5880–5901 (2022)
9. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282. PMLR (2017)
10. Mou, C., Zhang, J.: TransCL: transformer makes strong and flexible compressive learning. IEEE Trans. Pattern Anal. Mach. Intell. **45**(4), 5236–5251 (2023)
11. Nair, A., Liu, L., Rangamani, A., Chin, P., Bell, M.A.L., Tran, T.D.: Reconstruction-free deep convolutional neural networks for partially observed images. In: 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pp. 400–404. IEEE (2018)
12. Nasr, M., Shokri, R., Houmansadr, A.: Comprehensive privacy analysis of deep learning: passive and active white-box inference attacks against centralized and federated learning. In: 2019 IEEE Symposium on Security and Privacy (SP), pp. 739–753. IEEE (2019)
13. Park, J., Lim, H.: Privacy-preserving federated learning using homomorphic encryption. Appl. Sci. **12**(2), 734 (2022)
14. Sun, J., Li, A., Wang, B., Yang, H., Li, H., Chen, Y.: Soteria: provable defense against privacy leakage in federated learning from representation perspective. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9311–9319 (2021)
15. Truex, S., Liu, L., Chow, K.H., Gursoy, M.E., Wei, W.: LDP-fed: federated learning with local differential privacy. In: Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking, pp. 61–66 (2020)
16. Wang, Z., Song, M., Zhang, Z., Song, Y., Wang, Q., Qi, H.: Beyond inferring class representatives: user-level privacy leakage from federated learning. In: IEEE INFOCOM 2019-IEEE Conference on Computer Communications, pp. 2512–2520. IEEE (2019)
17. Wei, K., et al.: Federated learning with differential privacy: algorithms and performance analysis. IEEE Trans. Inf. Forensics Secur. **15**, 3454–3469 (2020)
18. Yin, H., Mallya, A., Vahdat, A., Alvarez, J.M., Kautz, J., Molchanov, P.: See through gradients: image batch recovery via gradinversion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16337–16346 (2021)
19. Zhao, B., Mopuri, K.R., Bilen, H.: iDLG: improved deep leakage from gradients. arXiv preprint arXiv:2001.02610 (2020)

20. Zhu, L., Liu, Z., Han, S.: Deep leakage from gradients. In: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Vancouver, BC, Canada, 8–14 December 2019, pp. 14747–14756 (2019)
21. Zisselman, E., Adler, A., Elad, M.: Compressed learning for image classification: a deep neural network approach. In: Handbook of Numerical Analysis, vol. 19, pp. 3–17. Elsevier (2018)